

INSTITUTO TECNOLÓGICO DE COSTA  
RICA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

PROGRAMA DE MAESTRÍA EN COMPUTACIÓN

Algoritmo cooperativo con enjambres de robots para  
la exploración de una zona y comunicación de una  
ruta hacia un objetivo

Tesis para optar al grado de Magister Scientiae en Computación

Rebeca Solís Ortega

Cartago, Costa Rica

Junio, 2016

# Resumen

En esta tesis se plantea el desarrollo, diseño y simulación computacional de un algoritmo cooperativo para ser usado por un enjambre de robots con el fin de explorar una zona y comunicar una ruta hacia un objetivo. Además se realiza una implementación física parcial del algoritmo con el fin de probar la viabilidad de ejecución del mismo.

El algoritmo realizado permite la organización independiente y descentralizada de los agentes que conforman el enjambre, esto mediante comunicación local y directa con el medio ambiente en que se desenvuelven.

Para el desarrollo del algoritmo se utilizó un esquema de huella virtual, mejor conocido como *feromonas*. Estas permiten a los agentes comunicarse con el medio y tomar decisiones con respecto a éstas, así los robots liberan una determinada feromona de acuerdo al estado en que se encuentren.

En general, se utilizaron “feromonas” de dos tipos: repelentes y atrayentes. Las repelentes, son liberadas por los agentes cuando éstos se encuentran en modo de búsqueda del objetivo e indican a los otros agentes zonas que ya han sido exploradas con anterioridad, por lo que generan en ellos un impulso para tomar un camino que no haya sido explorado. Por otro lado, las feromonas atrayentes se liberan una vez que el agente o ha llegado al objetivo o ha llegado a algún sector donde se encuentre dicha feromona, esto con el fin de comunicar la ruta hacia el objetivo.

El algoritmo propuesto, además de utilizar huellas virtuales, se basó en la teoría de autómatas celulares para determinar los estados y decisiones de los agentes de acuerdo al estado de su entorno inmediato (estado de los vecinos); esto con el fin de realizar una simulación computacional del mismo.

Finalmente, se realizó la construcción de enjambre de 5 robots físicos para poner a prueba el algoritmo base para la exploración de una zona en un ambiente a escala.

# Abstract

This thesis explains the development, design, simulation and physical implementation of a cooperative algorithm to be used by a swarm robotics in order to explore an area and communicate a route to a target. Furthermore a partial physical implementation of the algorithm is performed in order to test the feasibility of the implementation.

This algorithm allows independent and decentralized organization of a swarm of agents through local and direct communication with the environment in which they operate.

For the development of the algorithm a scheme that involves virtual footprint, better known as *pheromone* was used. These allow agents to communicate with the environment and take conscious decisions base on these foot marks.

Two types of pheromones was used: repellents and attractants. The repellents are released by the agents when they are in search mode and indicate to other agents, areas that have already been explored. On the other hand, attractant pheromones are released once the agent or has reached the goal or hit a sector where this pheromone is found, this in order to communicate the path to the target.

The algorithm is based on the theory of cellular automata to determine the states and decisions of the agents according to the state of their immediate environment.

Finally, a construction of 5 physical robots was made to test the basic algorithm for scanning an area on a scale environment.

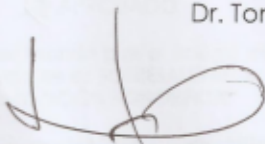
## APROBACIÓN DE LA TESIS

**"Algoritmo cooperativo con enjambres de robots para la exploración  
de una zona y comunicación de una ruta hacia un objetivo."**

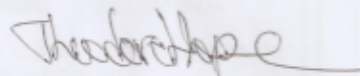
### TRIBUNAL EXAMINADOR



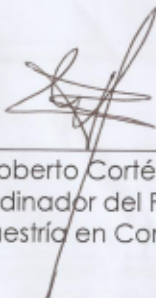
Dr. Tomás del Camino Beck.  
Profesor Asesor



Dr. José Castro Mora.  
Profesor Lector



MSc. Theodore Hope.  
Profesor Externo



Dr. Roberto Cortés Morales  
Coordinador del Programa  
de Maestría en Computación

Junio, 2016



# Dedicatoria

*A mi madre, por estar siempre a mi lado, apoyarme incondicionalmente y creer siempre en mí.*

*A mi padre, por cuidarme y ayudarme a que este título sea posible.*

*A Esmeralda, por ser la luz al final del túnel.*

*A Raquel y Karina por estar ahí.*

*A mamá Flor, por creer siempre en mí.*

# Agradecimientos

A Tomás por todo su apoyo y guía, sin él esta investigación no sería posible.

A Galina por ayudarme en todo el proceso, más de lo que se imagina.

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Definición del problema . . . . .	3
1.2. Justificación del problema . . . . .	3
1.3. Objetivos . . . . .	4
1.3.1. Objetivo general . . . . .	4
1.3.2. Objetivos específicos . . . . .	5
1.4. Alcance . . . . .	5
<b>2. Marco teórico</b>	<b>6</b>
2.1. Enjambre de robots . . . . .	6
2.1.1. Características . . . . .	6
2.1.2. Taxonomía . . . . .	11
2.1.3. Ventajas de los enjambres de robots . . . . .	13
2.1.4. Enjambre de robots y algoritmos cooperativos . . . . .	15
2.2. Autómatas celulares . . . . .	16
2.3. Uso de feromonas o “huellas virtuales” para la comunicación . . . . .	19
2.4. Trabajos relacionados . . . . .	21
2.4.1. Simulación computacional . . . . .	21
2.4.2. Implementaciones físicas . . . . .	23
<b>3. Entorno de trabajo del sistema de enjambres</b>	<b>27</b>
3.1. Medio . . . . .	27
3.1.1. Entorno o espacio . . . . .	28
3.1.2. Vecindario . . . . .	28
3.2. Agentes . . . . .	30
3.2.1. Características de los agentes . . . . .	30
3.2.2. Algoritmo de movimiento . . . . .	32

3.3. Ambiente computacional para la simulación del algoritmo . . . . .	36
<b>4. Algoritmo cooperativo con enjambres de robots para la exploración de una zona en búsqueda de un objetivo</b>	<b>41</b>
4.1. Algoritmo CA: Caminado aleatorio para la búsqueda de un objetivo . .	41
4.2. Algoritmo CAFRS: Caminado aleatorio para la búsqueda de un objetivo usando feromonas repelentes . . . . .	43
4.3. Pruebas experimentales del algoritmo CAFRS . . . . .	46
4.4. Algoritmo CAFRM: Modificación al algoritmo de caminado aleatorio usando feromonas repelentes . . . . .	52
4.5. Pruebas experimentales del algoritmo CAFRM . . . . .	55
4.6. Análisis de resultados de los algoritmos cooperativos con enjambres de robots para la exploración de una zona en búsqueda de un objetivo . .	57
<b>5. Algoritmo cooperativo con enjambres de robots para la comunicación de una ruta hacia un objetivo</b>	<b>60</b>
5.1. Algoritmo COFAS: Comunicación de un objetivo usando feromonas atra-yentes . . . . .	60
5.2. Pruebas experimentales del algoritmo COFAS . . . . .	63
5.3. Algoritmo COFAM: Comunicación de un objetivo usando feromonas atra-yentes modificado . . . . .	65
5.4. Análisis de resultados de los algoritmos cooperativos con enjambres de robots para la comunicación de una ruta hacia un objetivo . . . . .	67
<b>6. Algoritmo cooperativo con enjambre de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo</b>	<b>70</b>
6.1. Análisis de resultados del algoritmo cooperativo con enjambre de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo . . . . .	73
<b>7. Implementación física de un algoritmo propuesto</b>	<b>75</b>
7.1. Requerimientos mínimos de hardware para la implementación . . . . .	75

7.2. Análisis de resultados obtenidos por la implementación física . . . . .	81
<b>8. Conclusiones, recomendaciones y trabajo futuro</b>	<b>85</b>
8.1. Conclusiones . . . . .	85
8.2. Recomendaciones . . . . .	88
8.3. Trabajo futuro . . . . .	89
<b>9. Bibliografía</b>	<b>90</b>
<b>A. Anexos</b>	<b>94</b>
A.1. Resultados del algoritmo de caminado aleatorio con 1 agente . . . . .	94
A.2. Resultados del algoritmo de caminado aleatorio con 3 agentes . . . . .	95
A.3. Resultados del algoritmo de caminado aleatorio con 5 agentes . . . . .	96
A.4. Resultados del algoritmo de caminado aleatorio con 10 agentes . . . . .	97
A.5. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 1 agente . . . . .	98
A.6. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 3 agentes . . . . .	99
A.7. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 5 agentes . . . . .	100
A.8. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 10 agentes . . . . .	101
A.9. Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 1 agente . . . . .	102
A.10. Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 3 agentes . . . . .	103
A.11. Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 5 agentes . . . . .	104
A.12. Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 10 agentes . . . . .	105

# Índice de figuras

1.	Características de los enjambres de robots. . . . .	7
2.	Vecindarios de un autómeta celular de 2-dimensiones . . . . .	17
3.	Resultados obtenidos por Fossum utilizando feromonas repelentes para la exploración de una zona . . . . .	22
4.	Ejemplo de un sistema de comunicación a través de feromonas . . . . .	23
5.	Robot scout para explorar zonas luego de un desastre . . . . .	24
6.	Swarm-bots realizando tareas . . . . .	25
7.	Partes de un swarmanoid . . . . .	25
8.	Enjambre de Kilobots . . . . .	26
9.	Auto-organización de un enjambre de kilobots para lograr crear formas bidimensionales . . . . .	26
10.	Composición del medio de trabajo del sistema de enjambres . . . . .	27
11.	Definición del vecindario del espacio celular en el que se trabajará . . . . .	28
12.	Composición de los agentes del sistema de enjambres . . . . .	30
13.	Simbología de las funciones de transición de movimiento de un agente . . . . .	35
14.	Ejemplos de posibles estados de transición de movimiento . . . . .	35
15.	Entorno computacional para las simulaciones . . . . .	38
16.	Representación del tipo de feromonas . . . . .	40
17.	Esquema del algoritmo CA: Caminado aleatorio para la búsqueda de un objetivo . . . . .	43
18.	Esquema del algoritmo CAFRS: Caminado aleatorio para la búsqueda de un objetivo usando feromonas repelentes . . . . .	46
19.	Resumen estadístico de la ejecución del algoritmo de caminado aleatorio . . . . .	48
20.	Resumen estadístico de la ejecución del algoritmo de caminado aleatorio con feromonas repelentes . . . . .	49
21.	Comparación de datos estadísticos de los modelos de caminado aleatorio simple y caminado aleatorio con feromonas repelentes . . . . .	50

22.	Comparación general de los modelos de caminado aleatorio simple y caminado aleatorio con feromonas repelentes . . . . .	51
23.	Recorrido de los agentes en los modelos de caminado aleatorio simple y caminado aleatorio con feromonas repelentes . . . . .	52
24.	Esquema del algoritmo CAFRM: Modificación al algoritmo de caminado aleatorio usando feromonas repelentes . . . . .	53
25.	Resumen estadístico de la ejecución del algoritmo modificado de caminado aleatorio con feromonas repelentes . . . . .	55
26.	Comparación de datos estadísticos de los modelos de caminado aleatorio simple, caminado aleatorio con feromonas repelentes y modificación del caminado aleatorio con feromonas repelentes . . . . .	56
27.	Recorrido de los agentes en los modelos de caminado aleatorio con feromonas repelentes simple y modificado . . . . .	57
28.	Comparación métodos de caminado aleatorio con feromonas repelentes simple y modificado . . . . .	59
29.	Esquema del algoritmo COFAS: Comunicación de un objetivo usando feromonas atrayentes . . . . .	61
30.	Comparación del algoritmo base versus el algoritmo de comunicación utilizando feromonas atrayentes . . . . .	65
31.	Esquema del algoritmo COFAM: Comunicación de un objetivo usando feromonas atrayentes modificado . . . . .	66
32.	Comparación del algoritmo base, el algoritmo de comunicación utilizando feromonas atrayentes y su versión modificada . . . . .	68
33.	Resumen estadístico de la ejecución del algoritmo modificado con comunicación por medio de feromonas atrayentes de acuerdo a la cantidad de pasos . . . . .	69
34.	Esquema del algoritmo AECO: cooperativo con enjambre de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo	71
35.	Enjambre físico de robots . . . . .	76
36.	Ejemplo de un Pro Trinket 5V . . . . .	77
37.	Ejemplo de un sensor de color TCS34725 . . . . .	78

38.	Ejemplo de un NeoPixel Flora . . . . .	78
39.	Ejemplo de motores mini RC Quadcopter . . . . .	78
40.	Modelado de la estructura 3d para la montura de los robots . . . . .	79
41.	Ejemplo de un robot físico . . . . .	80
42.	Simulación espacio físico . . . . .	80
43.	Implementación física del algoritmo . . . . .	84
A.1.	Resultados del algoritmo de caminado aleatorio con 1 agente . . . . .	94
A.2.	Resultados del algoritmo de caminado aleatorio con 3 agentes . . . . .	95
A.3.	Resultados del algoritmo de caminado aleatorio con 5 agentes . . . . .	96
A.4.	Resultados del algoritmo de caminado aleatorio con 10 agentes . . . . .	97
A.5.	Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 1 agente . . . . .	98
A.6.	Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 3 agentes . . . . .	99
A.7.	Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 5 agentes . . . . .	100
A.8.	Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 10 agentes . . . . .	101
A.9.	Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 1 agente . . . . .	102
A.10.	Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 3 agentes . . . . .	103
A.11.	Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 5 agentes . . . . .	104
A.12.	Resultados del algoritmo modificado de caminado aleatorio con feromo- nas repelentes con 10 agentes . . . . .	105



## Lista de algoritmos

1.	<b>CA:</b> Caminado aleatorio para la búsqueda de un objetivo . . . . .	42
2.	<b>CAFRS:</b> Caminado aleatorio para la búsqueda de un objetivo usando feromonas repelentes . . . . .	45
3.	<b>CAFRM:</b> Modificación al algoritmo de caminado aleatorio usando fe- romonas repelentes . . . . .	54
4.	<b>COFAS:</b> Comunicación de un objetivo usando feromonas atrayentes .	63
5.	<b>COFAM:</b> Comunicación de un objetivo usando feromonas atrayentes modificado . . . . .	67
6.	<b>AECO:</b> Algoritmo cooperativo con enjambre de robots para la explora- ción de una zona y comunicación de una ruta hacia un objetivo . . . .	72

# 1. Introducción

Durante los últimos años, los sistemas robóticos móviles involucran el uso de un robot individual que opera dentro de un determinado ambiente para llevar a cabo ciertas tareas como búsqueda de objetos, mapeo de zonas, entre otros. Sin embargo, algunos investigadores como Dudek et al. [1], Sharkey y Sharkey [2], Ying y Zhong-Yang [3], consideran más eficiente utilizar una gran cantidad de robots simples y pequeños, con el fin de minimizar costos, optimizar la ejecución de tareas, tener un sistema tolerable a fallos, que sea flexible y fácilmente escalable. A este tipo de sistemas robóticos se le conoce como enjambres de robots.

Los enjambres de robots son un nuevo enfoque dirigido a la coordinación de los sistemas multi-robot con el fin de que emerja un comportamiento colectivo complejo, de la simple interacción entre los robots y el medio ambiente en el que se encuentran [3].

Los enjambres tienen un gran potencial para resolver problemas como lo son: control de tráfico, formaciones en movimiento, exploración, mapeo, búsqueda y rescate, simulación de escenarios y comportamientos biológicos, planificación de rutas, entre otros [2], [4], [5].

Estas tareas pueden ser ejecutadas, tanto por robots individuales o en grupo, sin embargo para que estos últimos, conocidos como enjambres, sean más eficientes que los primeros, diversos autores [1], [2], [3], [4] recomiendan que los agentes que conforman el enjambre cuenten con ciertas características como lo son:

- Simplicidad.
- Autonomía.
- Comunicación.
- Inspiración biológica.
- Control del sistema, que permita el paralelismo, la escalabilidad, estabilidad, flexibilidad y eficiencia.

Debido a estas características, los algoritmos cooperativos son los más utilizados para su programación. Se han estudiado y aplicado varios enfoques como por ejemplo lógica

difusa ([5], [6]), enjambre de partículas ([5]), modelo de máquina de soporte vectorial ([7]), métodos tradicionales (como A\* y *greedy*) y probabilísticos ([5], [8]), métodos basados en solución de laberintos ([9], [10]), la teoría de autómatas celulares ([11], [12], [13], [14], [15], [16], [17]), entre otros.

El presente documento propone una investigación del estado del arte de enjambres de robots, teoría de autómatas celulares, técnicas y algoritmos para explorar zonas desconocidas y otros más, con el fin de crear un algoritmo cooperativo que permita la solución del problema de exploración de una zona y comunicación de una ruta de salida en ambientes desconocidos haciendo uso de un enjambre de robots.

Dicho algoritmo se ejecutará mediante simulaciones computacionales con el fin de analizar los resultados que se obtengan. Además se realizará una implementación física de dicho algoritmo usando microcontroladores y componentes de bajo costo.

La distribución de los contenidos de esta tesis es la siguiente: en la sección 1 se expondrá la definición y justificación del problema que se desea resolver, además de enunciar la hipótesis de investigación y los objetivos relacionados a la misma. En la sección 2 se presenta el marco teórico, donde se realizará un estado del arte sobre los temas que engloba esta investigación, para esto se iniciará con los fundamentos teóricos de los enjambres de robots, la teoría de los autómatas celulares, el uso de feromonas por diversos individuos para lograr una mejor interacción con el medio en que se encuentran y finalmente se mostrarán algunos trabajos relacionados. En la sección 3 se definirá el entorno de trabajo del sistema multirobot. Luego en la sección 4 se expondrán los algoritmos propuestos sobre la búsqueda de un objetivo en una zona desconocida. En la sección 5 se enuncian los algoritmos propuestos con relación a la comunicación de una ruta hacia un objetivo en un terreno desconocido. En la sección 6 se realiza una unión de los dos algoritmos propuestos, con el fin de determinar un algoritmo eficiente (en cuanto a tiempo de duración) que permite explorar una zona y comunicar una ruta hacia un objetivo por medio de enjambres. Luego en la sección 7 se mostrarán los resultados obtenidos de la implementación física, para esto primero se analizarán los requerimientos de hardware con el que deben contar los robots para llevar a cabo dicho algoritmo para luego realizar el análisis respectivo. Finalmente en la sección 8 se presentarán las conclusiones, recomendaciones y el trabajo futuro que resulte de esta investigación.

## 1.1. Definición del problema

El problema a resolver en esta tesis es la exploración eficiente (en cuanto a tiempo de duración) de una zona y comunicación de una ruta hacia un objetivo por parte de un enjambre de robots, haciendo uso de algoritmos cooperativos centrados en la teoría de autómatas celulares.

Marcel et al. [18] nos indica que la tarea de cubrir de manera óptima un ambiente desconocido usando agentes “inteligentes” es un problema NP-Hard (o NP-complejo), debido a que no puede ser resuelto por medio exhaustivos, por lo que deben aplicarse métodos aproximados.

Debido a esto autores en [18], [19], [20] utilizan enfoques aproximados, lo cuales resultan más prácticos debido a su capacidad de proporcionar soluciones casi óptimas en un corto período de tiempo corto y sin necesidad de procesar todo el espacio de búsqueda (como sucede con los métodos exhaustivos).

Dentro del ámbito de enjambres de robots, el desarrollo de un algoritmo rápido y eficaz que le permita al grupo explorar una zona y comunicar una ruta hacia un objetivo, mediante la cooperación entre los agentes que lo componen y su comunicación con el medio, es un paso clave para el avance de la investigación de robots móviles ([7], [11]) y sus aplicaciones a futuro.

## 1.2. Justificación del problema

El tema de los enjambres de robots es relativamente nuevo y ha sido hasta en los últimos años, cuando se ha dedicado más a la investigación y creación de algoritmos que le permitan al enjambre realizar una cierta cantidad de tareas [2], [21], [22], [23]. En cuanto al problema de exploración y comunicación de una ruta se han propuesto varios métodos para resolverlo utilizando robots individuales o múltiples. Sin embargo la complejidad de algunos sistemas se incrementa de acuerdo con el número de robots en el enjambre, por eso la importancia de crear sistemas paralelos, escalables, flexibles y tolerantes a fallas [2], [3], [4]. Así mientras que los resultados de simulación muestran la eficacia del método, su implementación en un sistema real es una tarea compleja porque impli-

ca, en algunos casos, un control centralizado y robots más costosos y complejos [12], [13].

La importancia fundamental para resolver de manera eficiente el problema en cuestión con un enjambre de robots físicos, traerá consigo grandes avances en el área de búsqueda y rescate, pues se podrán utilizar enjambres para realizar exploraciones colectivas en ambientes peligrosos, con el fin de localizar y converger en un objetivo conocido [5]. Además, en la literatura consultada, no se ha realizado implementación física de un enjambre de robots que les permita, de manera descentralizada y cooperativa, encontrar una ruta hacia una cierta locación haciendo uso de la teoría de autómatas celulares en ambientes con gran cantidad de obstáculos empleando un enfoque con feromonas.

Otra aplicación importante es encontrar una salida en caso de derrumbes de minas, edificios u otros, donde la búsqueda de una salida segura es de suma importancia para evitar pérdidas humanas. En estos casos, factores como la visibilidad, sentido de orientación, sentido común y demás se ven significativamente reducidos en las personas que se encuentren atrapadas, y si a eso se le agrega la posibilidad de heridas físicas, resulta muy difícil para un individuo o grupo de personas encontrar la salida del lugar donde se encuentran. En casos como estos un enjambre de robots pueden ser de mucha utilidad para encontrar una salida y ayudar así a las personas a salir del lugar.

Así en esta tesis se exploran varios algoritmos para sistemas robóticos móviles basados en enjambres de robots, siguiendo un modelo bio-inspirado, con el fin de lograr un algoritmo cooperativo basado en la teoría de autómatas celulares, para explorar una zona, encontrar una salida y comunicar una ruta hacia esta de manera eficiente en cuanto al tiempo.

## **1.3. Objetivos**

### **1.3.1. Objetivo general**

Diseñar e implementar un algoritmo cooperativo usando enjambre de robots para la eficiente exploración de una zona y comunicación de una ruta hacia un objetivo.

### 1.3.2. Objetivos específicos

1. Diseñar un algoritmo cooperativo que permita la exploración eficaz de una zona desconocida.
2. Diseñar un algoritmo cooperativo que permita la comunicación de una ruta hacia una salida u objetivo.
3. Diseñar un algoritmo cooperativo que integre los algoritmos propuestos con el fin de explorar una zona y comunicar una ruta hacia una salida y objetivo.
4. Realizar una simulación de un enjambre de robots que ejecuten el algoritmo diseñado.
5. Determinar los requerimientos mínimos de hardware que debe tener un enjambre de robots para implementar de manera física el algoritmo diseñado.
6. Crear un grupo de robots con las características anteriormente señaladas.
7. Realizar una simulación física de un enjambre de robots que ejecuten alguno de los algoritmos diseñados.

### 1.4. Alcance

La tesis abarca elementos de análisis, diseño, implementación, pruebas y evaluación de los resultados. Para poder resolver el problema propuesto, se trabajará en ambientes, a escala, similares a laberintos, donde los agentes que componen el enjambre, se encontrarán situados en una posición arbitraria y desconocen tanto el entorno en que se ubican como la locación objetivo a la que deben llegar.

Los robots podrán comunicarse entre sí únicamente de manera local y podrán hacer uso de sensores que cubren un cierto radio alrededor de ellos. Además tendrán, relativamente, poca capacidad de memoria, por lo que las comunicaciones entre los individuos y el medio ambiente se harán por medio del uso de “huellas o marcas digitales” (feromonas digitales [24]).

## **2. Marco teórico**

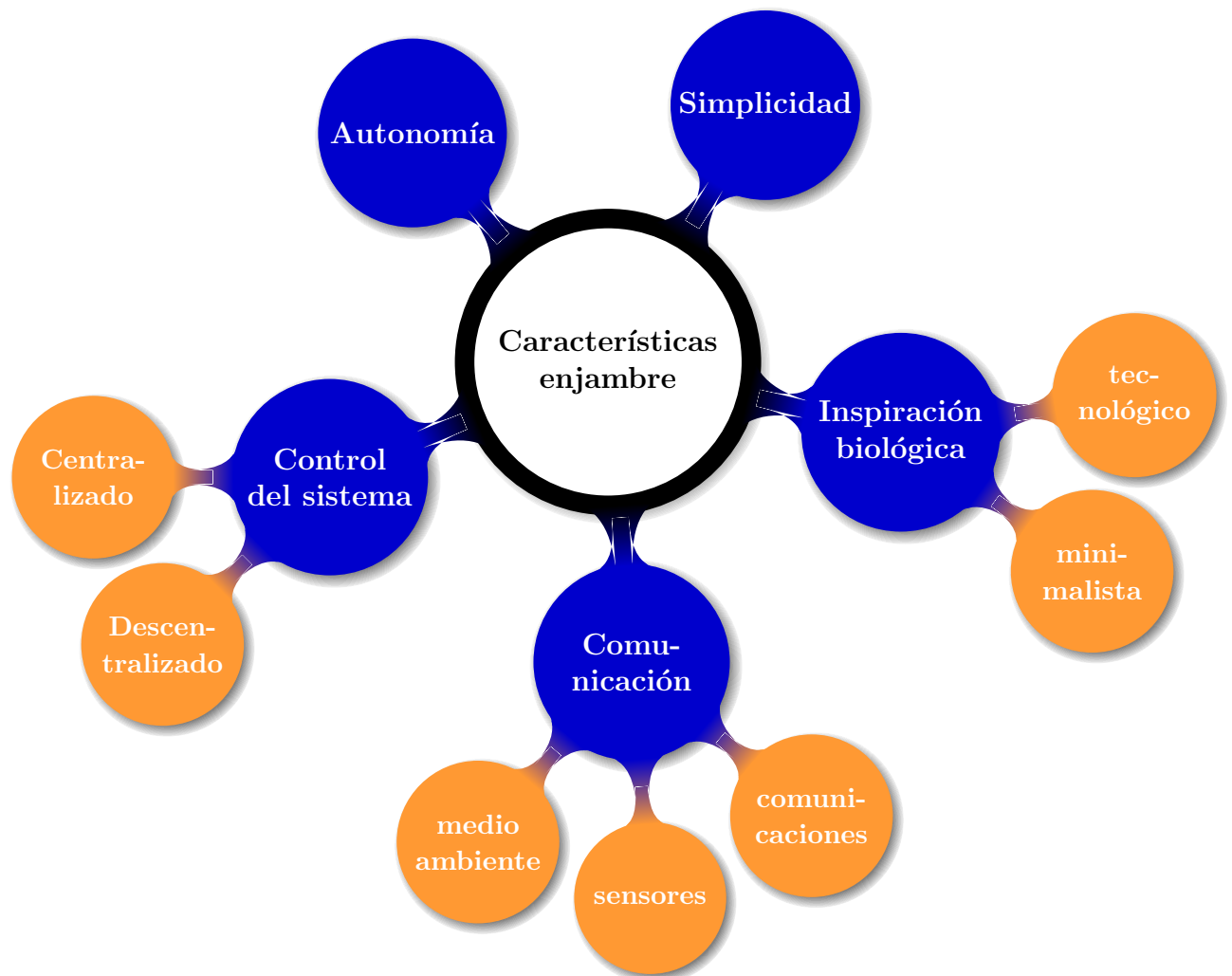
### **2.1. Enjambre de robots**

Los enjambres de robot tienen su origen en el concepto de inteligencia de enjambres, el cuál fue introducido por Beni y Wang en 1989 en su investigación sobre agentes simulados auto-organizables en el contexto de sistemas robóticos celulares [2]. La inteligencia de enjambres consiste en un énfasis biológicamente inspirado, donde el control descentralizado, la comunicación local propician la aparición de un comportamiento global como resultado de la auto-organización de los individuos que lo componen, ejemplo de esto son los comportamientos presentados por las hormigas, termitas, abejas, aves, peces, lobos y otros animales similares [4]. Así, en su definición más sencilla, los enjambres de robots son la aplicación de la inteligencia de enjambres a los principios de los robots colaborativos.

Una definición más robusta es dada por Tan y Zheng en [3], en donde mencionan que los enjambres de robots son un nuevo enfoque dirigido a la coordinación de los sistemas multi-robot compuestos por un gran número de robots físicos, en su mayoría simples. A partir de estos, y de ciertas reglas individuales, se espera que un comportamiento colectivo deseado emerja de la interacción entre dichos robots y la interacción de éstos con el medio ambiente en el que se encuentran.

#### **2.1.1. Características**

A pesar de las diversas definiciones sobre lo que es y representa un enjambre de robots, existe un consenso general sobre las características que debe tener un grupo de robots para ser considerado un enjambre. A continuación, se presenta un compendio de dichas características según lo presentado en [2], [3], [4] (Ver Figura 1).



**Figura 1:** Características de los enjambres de robots.

1. **Simplicidad:** Se desea que los robots sean simples, en el sentido de como están contruidos y de las necesidades de hardware que requieren para su funcionamiento, pues esto les permitirá responder de manera rápida y flexible a los cambios del ambiente en que se encuentren [2].

Además es esta característica la que genera que los costos de producción sean significativamente menores que a los de construir un robot individual que realice todo el trabajo del enjambre.

2. **Autonomía:** Se busca que los robots sean seres autónomos, osea que no sean controlados central ni remotamente, por servidores o seres humanos. Aunque pueden existir miembros "líderes" en el enjambre que decidan las acciones generales de



otros individuos.

3. **Control del sistema:** El sistema bajo el cual trabaja el enjambre puede darse de dos maneras según [4]:

- *Centralizado:* consiste en un sistema donde se tiene un robot líder, el cuál está encargado de organizar el trabajo de los otros robots. Él es quien se encarga de tomar las decisiones de todo el equipo, mientras que los otros miembros del grupo actúan de acuerdo a sus direcciones.
- *Descentralizado:* trata de un sistema donde todos los robots son miembros iguales y cada uno toma decisiones propias basadas en las interacciones que tienen con el ambiente o con otros robots.

En general, un sistema completamente descentralizado trae consigo ventajas como la tolerancia a fallos, evita los cuellos de botella y ayuda con la escalabilidad y flexibilidad del sistema [2], [3].

4. **Comunicación:** La comunicación de los robots es fundamental para la organización y cumplimiento eficiente de la tarea que deben realizar. Debido a la usual restricción de hardware, el costo de los robots y a la inspiración biológica, se busca que los enjambres tengan un rango limitado para detectar y comunicarse con otros robots o con elementos del ambiente que los rodea. El uso de una comunicación global disminuirá la escalabilidad y la flexibilidad del sistema, dado que el costo de la comunicación aumenta exponencialmente a medida que la población crece [2], [3]. Sin embargo, en algunos modelos la comunicación global de control son aceptables, como por ejemplo, cuando se actualiza las estrategias de control o se da el envío de las señales de terminales, eso si siempre y cuando no se utilice en la interacción entre los individuos [3].

Se pueden encontrar tres tipos de formas en las cuales se puede dar la comunicación entre los diversos robots, entre estos encontramos:

- *Interacción a través de el medio ambiente:* se refiere a la forma más simple de comunicación y está directamente relacionada con la inspiración biológica de los enjambres. En este caso los robots realizan diversas actividades de acuerdo a los elementos que se encuentren en el ambiente [2].

En este tipo de interacción los robots pueden dejar señales en el ambiente después de una determinada acción, con el fin de estimular y comunicar a otros robots situaciones particulares, sin necesidad de tener una comunicación directa entre ellos, así las acciones posteriores tienden a ser reforzadas y de esta manera propiciar la aparición espontánea de las actitudes que se desea del enjambre [3].

Un ejemplo de este tipo de comunicación son las señales físicas (conocidas como feromonas virtuales) que pueden ir dejando los robots durante su recorrido, con el fin de que otros robots pueden procesarlos y así modificar su comportamiento [2], [3].

- *Interacción a través de sensores:* es la que resulta de la capacidad de los robots de detectarse entre sí, aunque no haya una comunicación explícita entre ellos [2]. Este tipo de interacción permite a los robots distinguir entre otros individuos del enjambre y objetos propios del ambiente con el fin de interactuar de diferente manera con ellos (como evasión de obstáculos, búsquedas de destinos, reuniones, etc.), así el principal objetivo de integrar sensores en el enjambre es la cooperación, tratando de simplificar la comunicación [3].
- *Interacción a través de comunicaciones:* acá la interacción le permite a los robots tener un conocimiento global de las actividades que están llevando a cabo los otros integrantes del enjambre. La comunicación en este tipo es similar a la de una red inalámbrica y generalmente se compone de dos tipos: *peer-to-peer* y *broadcast*, aunque con el desarrollo de dispositivos móviles otras tecnologías existentes también pueden adoptarse.

Este tipo de comunicación ha sido grandemente criticada por estar muy poco relacionada con la noción de inspiración biológica del sistema, además de producir dificultades de escalamiento, cuellos de botella, flexibilidad y au-

mentar las necesidades de hardware de los robots y por ende su costo [2].

Ha habido mucho debate sobre el nivel de comunicación que se debe permitir, unos investigadores opinan que la comunicación debe mantenerse a un nivel implícito a través del medio ambiente y/o sensores mientras que otros opinan que se debe dar explícitamente a través de comunicaciones [4]. Y si bien todavía no hay un consenso general, actualmente se incentiva a la comunidad científica a preferir un tipo de comunicación implícita por las características de robustez asociadas a este tipo.

5. **Inspiración biológica:** dado el origen de los enjambres de robots, se desea que su comportamiento esté, en cierto sentido, biológicamente inspirado. Animales que viven en sociedad, como las hormigas, termitas, abejas, aves, peces y otros, representan el mejor ejemplo de como seres con poca capacidad cognitiva pueden lograr labores impresionantes como lo es el mantenimiento de la salud de la colonia, cuidado de crías, búsqueda de comida en zonas desconocidas, respuesta a invasiones, y muchas otras más [4].

Existen un debate abierto sobre que tanto un enjambre debe estar relacionado con los comportamientos biológicos de los seres anteriormente expuestos, por lo que se han creado dos corrientes [2]:

- *Enfoque minimalista:* en el cual las características físicas y de comunicación de los robots deben limitarse a un nivel mínimo.
- *Enfoque tecnológico:* en el cual se incentiva a utilizar algunas de las capacidades especializadas ofrecidas por la electrónica moderna.

Cada una de estas corrientes traen consigo ventajas y desventajas, por lo que la elección del enfoque dependerá de los objetivos que se deseen alcanzar.

Dadas las características y ventajas que trae consigo el uso de enjambres de robot, se recomienda su uso en múltiples áreas. En general dichas aplicaciones se pueden clasifi-

car en dos clases [3]:

1. **Basados en patrones:** problemas donde los patrones son parte importante para la solución del mismo, como por ejemplo, la agregación, cartografía, migración, redes auto-organizadas, despliegue de agentes distribuidos, cobertura de áreas, etc.
2. **Basados en entidades en el entorno:** Por el contrario, estos problemas se centran en los elementos que se encuentran en el entorno, como por ejemplo, la búsqueda de objetivo, detección de fuentes de olor, rescate de las víctimas en zonas de desastre, entre otras.

A pesar de esta distinción en los tipos de problemas a abordar, existen modelos híbridos que incluyen ambas clases, ejemplo de esto es la cooperación para transporte de elementos, la remoción de minas, etc. Más adelante analizaremos algunas aplicaciones concretas que se han realizado en los últimos años.

### 2.1.2. Taxonomía

Además de las características anteriormente expuestas, Dudek, Jenkin, Milios y Wilkes [1] fueron uno de los primeros en proponer una taxonomía para los enjambres de robots. Dicha clasificación ha sido una de las más aceptadas y mencionadas por diversos autores como en [2]. Siguiendo el esquema expuesto en [1] se presenta a continuación una parte de dicha clasificación, que es la que resulta más relevante para esta tesis.

1. **Tamaño del enjambre:** corresponde a la cantidad de robots del enjambre que se encuentran en el ambiente. En la literatura se consideran cuatro tipos:
  - *Individual (ALONE):* es el caso en que se trabaje con un sólo robot y técnicamente no se reconoce como enjambre.

- *Par (PAIR)*: es el caso de enjambre más simple y relaciona el uso de sólo dos robots.
- *Grupo limitado (LIM-GROUP)*: consiste en un enjambre con una cantidad  $n$  de robots, donde el número  $n$  es relativamente pequeño comparado con el tamaño del ambiente en que se desarrolla la tarea.
- *Grupo infinito (INF-GROUP)*: trata de un enjambre con una gran cantidad de robots, donde el ambiente resulta pequeño comparado con la cantidad de individuos que se tiene.

2. **Rango de comunicación:** corresponde al alcance de los robots para comunicarse con otros individuos de su enjambre. En general pueden darse tres tipos:

- *Comunicación nula (COM-NONE)*: acá los robots no pueden comunicarse con otro robot de su grupo de manera directa. Puede comunicarse de manera indirecta detectando su presencia, ausencia o comportamiento. En general trata de robots que no poseen ningún tipo de conocimiento, por lo que no se considera un enfoque muy realista o útil.
- *Comunicación local (COM-NEAR)*: en este enfoque los robots pueden comunicarse únicamente con otros robots que se encuentren dentro de un rango específico.
- *Comunicación global (COM-INF)*: este modelo todos los robots se pueden comunicar con todos los integrantes del enjambre en cualquier ubicación que se encuentren. Este tipo de comunicación puede ser muy poco práctica ya que acarrea con sí problemas como cuellos de botella, pérdida de información y otros; además de limitar la simpleza de los robots.

3. **Reconfigurabilidad del enjambre:** consiste en la capacidad y rapidez en que el enjambre puede organizarse con respecto a los cambios de la topología, el ambiente o ciertos acontecimientos que puedan ocurrir. Esta característica está estrechamente relacionada con el rango de comunicación que presenta el grupo. En

este apartado se tienen tres posibilidades:

- *Estático (ARR-STATIC)*: en este caso los robots no se re-ordenan u organizan ante los cambios que pueden ocurrir. Esto produce un enjambre frágil y propenso a fallas.
- *Coordinado (ARR-COMM)*: se da una re-organización ligada a la comunicación local entre los robots vecinos.
- *Dinámico (ARR-DYN)*: la relación de los miembros del enjambre puede cambiar arbitrariamente.

4. **Composición del enjambre:** se relaciona con la composición física y la funcionalidad de los robots. Así tenemos dos tipos:

- *Homogéneo*: los individuos presentan las mismas características y funcionalidades.
- *Heterogéneo*: al contrario del tipo anterior, los individuos pueden diferir en sus características y funcionalidades, pero debe existir una cierta cantidad de robots que compartan los mismos roles, sino no se consideraría un enjambre. Un ejemplo de esto son los robots usados para jugar fútbol, los cuales no son considerados enjambres debido a que cada individuo tiene un rol único e independiente del resto [3].

### 2.1.3. Ventajas de los enjambres de robots

Hasta el momento hemos visto las características y tipos de enjambres de robots que existen, y se ha sugerido que su implementación es más conveniente que el de utilizar un único robot que es más complejo tanto en hardware como en software. A continuación se presentan algunas ventajas mencionadas principalmente por Tan y Zheng en [3] y también en [2], [4].

1. **Paralelismo:** Debido a que el tamaño de la población de un enjambre es bastante grande, los robots pueden dividirse para realizar múltiples tareas al mismo tiempo, caso contrario si se trabajara con un único robot, el cuál este debería hacer las tareas de manera secuencia. Esta característica de los enjambres produce un ahorro importante de tiempo.
2. **Escalabilidad:** Si se da una comunicación local dentro del enjambre, se permitirá que varios individuos se puedan unir o salir de las actividades que se estén realizando sin interrumpir el esquema general del enjambre. Esto le permite al grupo adaptarse al cambio en la población sin la necesidad de ninguna intervención externa, lo que será muy útil para su aplicación en la vida real.
3. **Estabilidad:** Esta ventaja es similar a la de escalabilidad, pues el sistema en general no se verá afectado en gran medida, si una parte del enjambre deja las actividades debido a ciertos factores (dejan de funcionar, son destruidos u otros). Así el enjambre seguirá trabajando para cumplir el objetivo propuesto aunque dicha actividad ponga en riesgo a algunos miembros del grupo. Esta característica es especialmente útil para las tareas en un ambiente peligroso.
4. **Económico:** Como se ha mencionado anteriormente el costo de crear robots para un enjambre es relativamente bajo tanto en su diseño como su fabricación y su mantenimiento. Así en general todo el sistema es más barato que un sólo robot de gran complejidad, aún cuando se elaboran cientos o miles de robots, pues estos pueden ser producidos de manera masiva mientras que un solo robot complejo requiere de maquinaria con gran precisión y costo.
5. **Eficiencia energética:** Debido a que los individuos de un enjambre de robots son mucho más pequeños y simples que un robot grande y complejo, su gasto de energía será mucho menor, debido al tamaño de la batería y de sus componentes. Por lo que el tiempo de vida del enjambre será en general mayor, además en ambientes donde no se cuente con electricidad los enjambres pueden ser mucho más útiles.

6. **Flexibilidad:** Un mismo enjambre puede realizar diferentes tareas con el mismo hardware con que cuentan o bien con pequeñas modificaciones del mismo. De igual manera el grupo de robots pueden cambiar estrategia de acuerdo al medio ambiente en que se encuentren para lograr el cumplimiento del objetivo de manera más eficiente.

#### 2.1.4. Enjambre de robots y algoritmos cooperativos

Dado que investigación sobre enjambres de robots es relativamente nueva, todavía no existe un algoritmo general aplicable a gran cantidad de problemas; sino que se diseñan para cada aplicación específica. Una razón es que todavía no hay una definición común y estándar para el sistema de enjambres y los problemas de aplicación, debido a esto el progreso de la dicha investigación todavía avanza de manera lenta [3].

Además los algoritmos que se programen para ser utilizados por enjambres deben concordar y aplicar las características de los robots, que se expusieron anteriormente. Por lo que en general el algoritmo que se programe debe explotar la cooperación entre los robots. Algunas características que deben tener éstos, para ser utilizados en un enjambre de robots (mencionadas por Tan y Zheng en [3]) corresponden a:

1. **Simple:** Con el fin de ayudar a reducir el costo de confección, por esto, en muchos casos, los robots son considerados como una máquina de estado finito con sólo unos pocos estados.
2. **Escalable:** Debe ser capaz de funcionar bajo cualquier tamaño de la población y que permita a los robots salir y entrar del sistema de manera dinámica sin afectar el funcionamiento global del enjambre.
3. **Descentralizado:** Dado que los robots en un enjambre son autónomos, el algoritmo también debe serlo. Debe permitir que los robots tomen decisiones “por su cuenta” a partir de datos que encuentren en el entorno o que les suministren otros robots.



4. **Local:** Para aprovechar el potencial de los enjambres la comunicación e interacción de los robots con otros robots y con elementos del ambiente debe ser local. Se debe evitar en la medida de lo posible, el uso de operaciones globales, pues podrían acarrear problemas como cuellos de botella, fallos en la comunicación y desincronización de los individuos.
5. **Paralelo:** Dado que los enjambres consisten de muchos robots, los algoritmos deben ser tan paralelos como sea posible, para que los robots pueden trabajar con múltiples tareas al mismo tiempo.

De acuerdo con estas características, se han propuesto una serie de algoritmos y enfoques para la creación de dichos algoritmos, algunos ejemplos de estos son lógica difusa [5], [6], enjambre de partículas [5], modelo de máquina de soporte vectorial [7], inclusive enfoques tradicionales como  $A^*$  y *greedy* en conjunto con métodos probabilísticos [5], [8] o bien métodos relacionados con la solución de laberintos [9], [10] y la teoría de autómatas celulares [11], [12], [13], [14], [15], [16], [17].

Sin embargo, por tratarse de un tema nuevo, los principales intereses de los investigadores son algunas de las tareas básicas, tales como el control de la formación, evitar obstáculos, entre otros, por lo que todavía se carece de un marco unificado [3].

## 2.2. Autómatas celulares

Como se puso observar en la sección anterior existen varios modelos o enfoques que se pueden utilizar para la creación de algoritmos cooperativos orientados a enjambres de robots. En esta tesis en particular, nos centraremos en la teoría de autómatas celulares, este enfoque se encuentra entre los modelos más antiguos de la computación natural y fue desarrollada por John Von Neumann en la década de los 40; su principal inspiración fue el comportamiento biológico de algunos seres y su objetivo principal fue diseñar sistemas artificial auto-replicable que fuera computacionalmente universal [25], [26].

Así Von Neuman, siguiendo las sugerencias de S. Ulam, creó un universo discreto que consiste en una malla bidimensional de máquinas de estados finitos a los cuales llamó

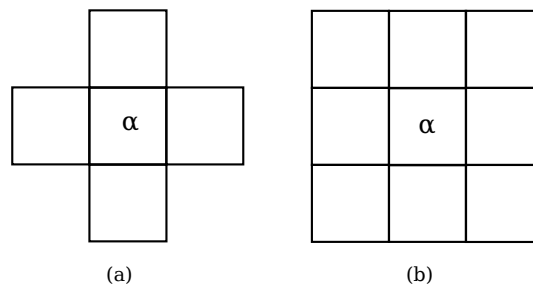
células, interconectadas a nivel local con los demás. Estas células cambian sus estados de forma sincrónica en función de los estados de algunas células vecinas según lo determinado por una regla de actualización local. Todas las células utilizan la misma regla de actualización. Hoy en día a estos universos celulares se les conoce como autómatas celulares [26].

Un espacio celular consiste en un espacio euclídeo infinito de  $n$ -dimensiones en donde existen relaciones definidas para un cierto vecindario. Dichas relaciones permiten a cada celda mantener una lista finita de celdas vecinas en un tiempo discreto definido  $t = 0, 1, 2, 3, \dots$  [25].

Así un sistema de autómatas celulares (o sistema celular) esta dado por [25]:

- Una lista finita de estados para cada celda.
- Un estado definido para cada celda.
- Una regla que define el estado de la celda en tiempo  $t + 1$  como función de su propio estado y el de sus vecinos en el tiempo  $t$ .

En la Figura 2 se muestran dos tipos de espacios celulares en dos dimensiones.



**Figura 2:** Vecindarios de un autómata celular de 2-dimensiones (a) Vecindario Von Neumann, con 5 celdas vecinas. (b) vecindario de Moore, con 9 celdas vecinas.

En términos matemáticos, Codd en [27] nos dice que, dado un conjunto  $I$  de números enteros, para obtener un espacio espacio celular se asocia con el conjunto  $I \times I$  con:

1. **Vecindario:** El cuál se define por medio de una función  $g : I \times I \rightarrow 2^{I \times I}$  tal que:

$$g(\alpha) = \{\alpha + \delta_1, \alpha + \delta_2, \dots, \alpha + \delta_n\}$$

$\forall \alpha \in I \times I$  donde  $\delta_i (i = 1, 2, \dots, n) \in I \times I$  son constantes.

2. **Autómata finito:** El cuál se define como  $(V, v_0, f)$  donde  $V$  es el conjunto de espacio celular,  $v_0$  es un elemeto de  $V$  llamada el estado inactivo o inicial, y  $f$  es la función local de transición de las  $n$ -tuplas de elementos de  $V$  en  $V$ . Dicha función  $f$  está sujeta a la restricción:

$$f(v_0, v_0, \dots, v_0) = v_0$$

Así podemos decir que la locación de cada celda está especificada por un sistema cartesiano de coordenadas con respecto a un origen y ejes arbitrarios, donde cada una de ellas contiene una copia idéntica del autómata finito  $(V, v_0, f)$ .

El estado  $v^t(\alpha)$  de una celda  $\alpha$  en el tiempo  $t$  es el estado asociado a dicho autómata en el tiempo  $t$  y cada una de ellas está conectada con  $n$  celdas vecina  $\alpha + \delta_1, \alpha + \delta_2, \dots, \alpha + \delta_n$ . En general se asume un vecino de una celda  $\alpha$  es ella misma, por lo que tendríamos que  $\delta_1 = 0$ .

Así la función de estado del vecindario viene dado por  $h^t : I \times I \rightarrow V^n$  definida como:

$$h^t(\alpha) = (v^t(\alpha), v^t(\alpha + \delta_2), \dots, v^t(\alpha + \delta_n), )$$

De este manera podemos relacionar el estado de los vecinos de una celda  $\alpha$  en el tiempo  $t$  en el espacio celular como la celda en el estado  $t + 1$  como:

$$f(h^t(\alpha)) = v^{t+1}(\alpha)$$

Dada su definición formal, pueden existir varios tipos de autómatas celulares de acuerdo a los posibles estados iniciales que se puedan presentar [25]:

1. **Determinístico:** Es aquel que tiene un único estado inicial y por lo tanto tendrá una única historia a través del tiempo.
2. **No determinístico:** El cuál puede tener más de un sólo estado inicial y su función de transición se basa en un conjunto de posibles siguientes estados para una celda.
3. **Probabilístico:** Este es un autómata celular no determinístico que posee una distribución de probabilidad sobre el estado inicial del autómata, y sobre los otros posibles estados de cada celda.

Los autómatas celulares poseen varias propiedades fundamentales del mundo físico: son paralelos, se manejan mediante interacciones homogéneas y todos son locales. Es por esto que muchos sistemas físicos y biológicos se han simulado con éxito utilizando este tipo de modelos [26]. Por esta característica no es de extrañar que varios investigadores hayan usado el enfoque de autómatas celulares y lo hayan aplicado en problemas relativos a enjambres de robots [11], [12], [13], [14], [15], [16], [17].

### 2.3. Uso de feromonas o “huellas virtuales” para la comunicación

Uno de los puntos focales cuando se trabaja con enjambres de robots, es el tipo de comunicación que existirá entre ellos y el ambiente. Si bien se pueden usar modelos centralizados en donde los robots sigan, en cierta medida, instrucciones de un servidor central o que utilicen diferentes técnicas de comunicación por medio de mensajes remotos, estos encarecen y complican dichos sistemas. Es por esto que en diversas investigaciones se recomiendan utilizar métodos inspirados en la biología [1],[2],[3],[5],[18], [28].

La naturaleza nos ofrece una gran cantidad de ideas y heurísticas que se pueden adaptar fácilmente a problemas de optimización. Marcel et al. en [18] afirman que el uso

de algoritmos bio-inspirados en enjambres de robots son pertinentes, sobre enfoques clásicos centralizados, cuando el espacio de la solución del problema es muy dinámico, impredecible, no-estructurado y no existe información previa sobre el medio ambiente. Si se analiza el problema de exploración de una zona, las características mencionadas anteriormente están presentes en éste, por lo que, un enfoque descentralizado distribuido para guiar el enjambre de robots para alcanzar una meta común proporciona robustez, escalabilidad y una mejor tolerancia a fallos sobre las técnicas centralizadas, características esenciales en un enjambre [2], [3], [4].

Estos mecanismos de control bio-inspirados solucionan de manera eficiente problemas que están muy por encima de las capacidades de un robot individual [28]. Cabe destacar que el éxito de un grupo robots en la realización de una tarea común se basa en las interacciones entre los miembros que conformen dicho sistema, por esto es de suma importancia analizar las técnicas y mecanismos de comunicación que utilizan las sociales de insectos y otros animales en la naturaleza, para poder elegir el modelo más adecuado.

Una de las técnicas que sobresalen en este tipo de algoritmos son las relacionadas con la estigmergia, el cual consiste es una forma particular de comunicación indirecta utilizada por los insectos sociales para coordinar sus actividades. Al explotar este enfoque en la coordinación, los investigadores han sido capaces de diseñar una serie de algoritmos exitosos en diversos campos de aplicación como los son la optimización combinatoria, el enrutamiento en redes de comunicación, la asignación de tareas en un sistema multi-robot, el análisis exploratorio de datos, dibujo gráfico y otros [29].

Un ejemplo del uso de la estigmergia en la naturaleza son las utilizadas por las hormigas y termitas. Estos grupos de individuos utilizan lo que se conoce como feromonas para comunicarse con el ambiente en que se desarrollan.

Las feromonas son sustancias químicas utilizadas en la naturaleza como una forma de comunicación indirecta que permite la realización de actividades organizadas en un grupo [30]. Por ejemplo, las hormigas dejan un rastro de feromonas para marcar el camino que atraviesan entre su nido y una fuente de alimento, así a medida que más hormigas recorren este camino, el rastro de feromona se refuerza lo que permite una mejor organización y comunicación en el grupo [2], [13], [29], [30].

Las feromonas tienen características importantes entre las que destacan su capacidad de evaporación y difusión. La evaporación determina la velocidad en la que disminuye la intensidad de la señal, mientras que la difusión se refiere a la rapidez con que las feromonas se propagan a través del ambiente [19]. Estas características le permiten a las feromonas jugar dos roles fundamentales las atrayentes, que propician la convergencia de individuos en zonas específicas, y las repelentes que por el contrario, propician que los individuos se alejen de ciertas zonas.

Dentro del ámbito de la computación, se pueden encontrar diversos autores que han creado algoritmos interesantes utilizando estas señales biológicas [18], [19], [28], [30].

## **2.4. Trabajos relacionados**

Dado que el tema de enjambres de robots es relativamente nuevo y hay mucho por investigar; existen varios proyectos e investigaciones donde se centran en un problema en específico (auto-organización, búsqueda de comida, rutas óptimas, planificación de rutas, mapeo, etc.). En esta sección nos centraremos en investigaciones de dos tipos:

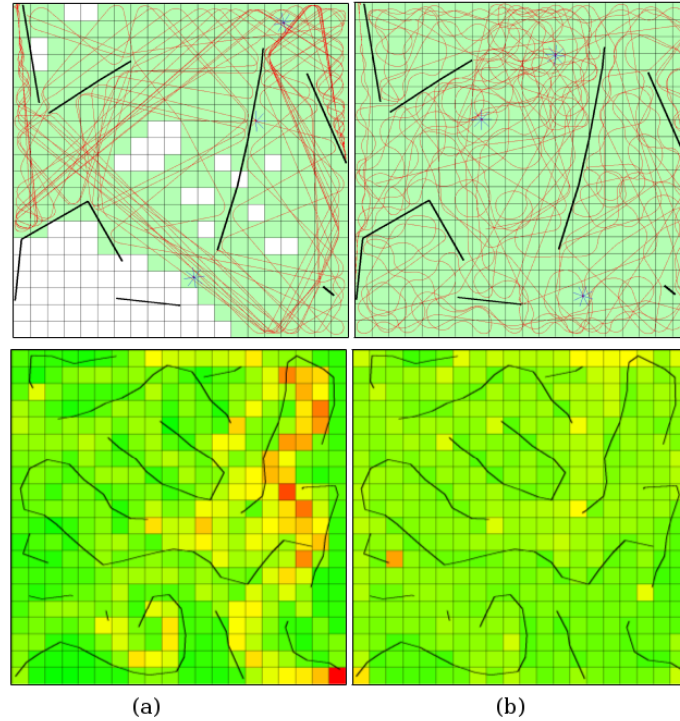
- Simulación computacional: aquellas en donde se proponen algoritmos relacionados con exploración de zonas o planificación de rutas por parte de enjambres simulados de robots.
- Implementaciones físicas: aquellas en donde se han construido enjambres físicos y hayan puesto a prueba sus algoritmos en ambientes reales (en ocasiones a escala).

### **2.4.1. Simulación computacional**

Fossum [19] explora el uso de feromonas repelentes en un entorno estático con el fin de conseguir una dispersión eficiente de un enjambre de robots; éstos no poseen ningún conocimiento previo de los entornos ni la locación precisa de los otros robots del enjambre. Para lograr una dispersión efectiva hace uso de feromonas virtuales repelentes para que funcionen como una memoria colectiva de las zonas visitadas anteriormente, destinadas a prevenir que varios agentes exploren los mismos lugares dos veces, aumentando así la

eficiencia de exploración.

Los resultados experimentales mostraron que las feromonas repelentes pueden ayudar a los robots en la exploración de entornos desconocidos de manera más eficiente y consistente (Ver Figura 3). Además se demuestra que los beneficios del uso de feromonas es mayor en enjambres pocos individuos y/o cuando los ambientes son grandes o contienen pocos obstáculos.



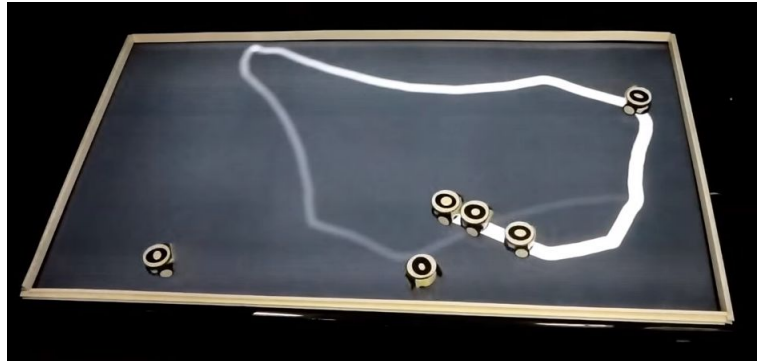
**Figura 3:** Resultados obtenidos por Fossum para la exploración de una zona (tomado de [19]) (a) Sin feromonas (b) Usando feromonas repelentes

Marcel et al. en [18] presentan un algoritmo para la cobertura espacial óptima de una región desconocida por un enjambre de agentes. El algoritmo se basa en la heurística de optimización de colonia de hormigas, en donde cada agente deja un rastro de feromonas virtual durante su movimiento a través de la región desconocida; este rastro puede ser atractor o repelente.

### 2.4.2. Implementaciones físicas

#### Colias

Colias es un proyecto realizado en la Universidad de Lincoln en el Reino Unido y está dirigido por Farshad y Shigang [28]. Dicho proyecto consiste en introducir un sistema de comunicación a través de feromonas donde, a través de sensores de luz y de una pantalla LCD para simular el ambiente, los robots son capaces de decodificar y seguir el rastro de una feromona (Ver Figura 4).



**Figura 4:** sistema de comunicación a través de feromonas (tomado de [28])

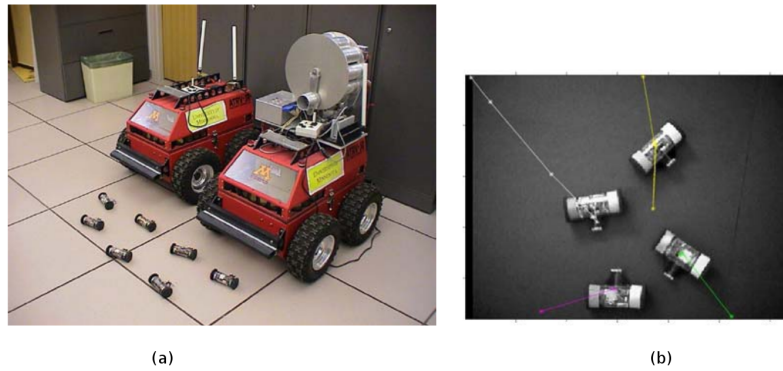
Este proyecto está actualmente activo y la creación de este sistema consiste en una primera etapa que busca llegar a crear aplicaciones y algoritmos que permitan resolver varios problemas de exploración y comunicación [28].

#### Scouts robots

Este proyecto publicado en el 2006, fue realizado por las universidades de Berea y Minnesota en Estados Unidos y consiste en utilizar el concepto de feromonas virtuales repelentes para lograr un algoritmo robusto que permite la dispersión de un equipo robótico centralizado [30].

El objetivo principal de esta investigación era de contribuir en la búsqueda y rescate de personas que hayan estado en zonas de desastres. Para esto utilizaron un modelo que consiste de dos robots: uno principal que funciona como un observador externo que permite proporcionar estimaciones de las posiciones de los minirobot (llamados scouts) quienes son los encargados de mapear una zona en busca de sobrevivientes (Ver Figura 5).





**Figura 5:** Robot scout para explorar zonas luego de un desastre (tomado de [30]) (a) Robot principal (b) Robots scouts

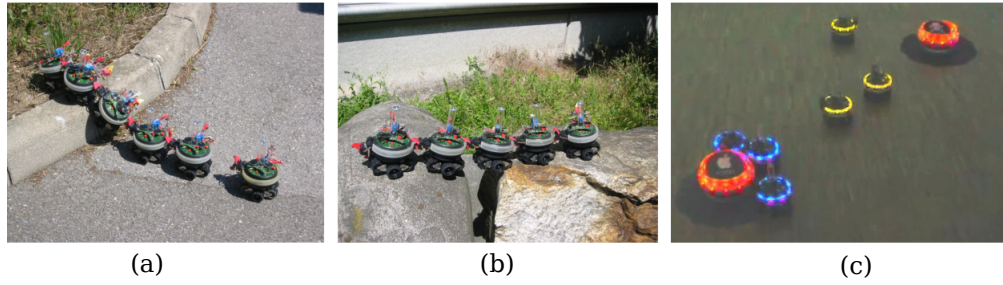
### SwarmBots project

Swarm-bots fue un proyecto patrocinado por el programa de la Comisión Europea de Tecnologías Futuras y Emergentes (FET-OPEN), inició el 1 de Octubre del 2001 y finalizó con éxito el 31 de Marzo del 2005 [31].

El trabajo llevado a cabo se centra en el estudio de los mecanismos que rigen los procesos de auto-organización y auto-ensamblaje de en agentes autónomos artificiales [21]. Para esto se crearon unos robots que llamaron swarm-bot, basados en la teoría de los enjambres de robots. Dichos robots son robots móviles con la capacidad de conectarse y desconectarse unos de otros. Sus vínculos físicos se utilizan para resolver problemas que no pueden ser solucionados por un solo swarm-bot, como por ejemplo: formar cadenas de tracción para recuperar un objeto pesado, navegación en terrenos difíciles donde los vínculos físicos pueden servir de apoyo si el swarm-bot tiene que pasar por encima de un agujero más grande que su tamaño, o cuando se tiene que pasar a través de una región cóncava empinada (Ver Figura 6) [21].

### Swarmanoid project

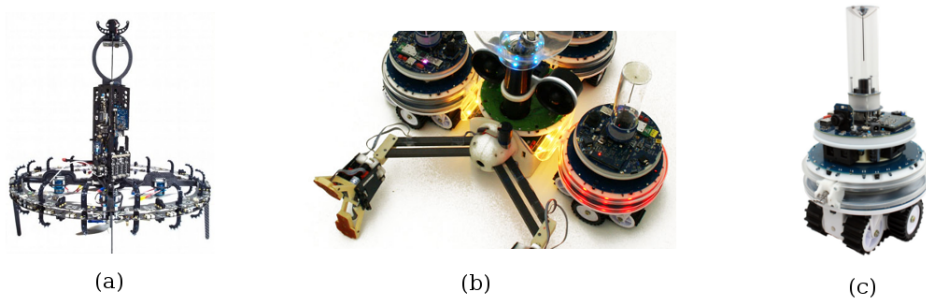
El proyecto Swarmanoid fue un proyecto financiado por la Comisión Europea Tecnologías Futuras y Emergentes (FET-OPEN), cuyo principal objetivo es la investigación del diseño, la ejecución y el control de una nuevo sistema robótico distribuido. Inició el 1 de Octubre del 2006 y finalizó el 30 de setiembre de 2010 (aunque desde entonces



**Figura 6:** Imágenes de swarm-bots realizando tareas (tomado de [21]). (a) Desplazándose cooperativamente en un ambiente irregular (b) Cruzando una zona donde hay un agujero más grande que el tamaño de un individuo (c) Buscando y transportando un elemento a un lugar específico.

se han hecho experimentos y algoritmos basados en dichos robots), surgió como una ampliación del trabajo realizado en el proyecto Swarm-bots. [3].

El sistema creado se compone de 60 pequeños robots autónomos y heterogéneos que se conectan de forma dinámica, dichos robots con tres tipos eye-bots, hand-bots y foot-bots (Ver Figura 7). Los tres robots en conjunto forman lo que los investigadores llamaron un swarmanoid. El objetivo principal del proyecto no era solamente crear dichos robots sino también el de desarrollar algoritmos distribuidos para el control de los swarmanoid, además de estudiar y definir protocolos de comunicación distribuidos que le permitan al sistema trabajar de manera distribuida, robusta y escalable [23].

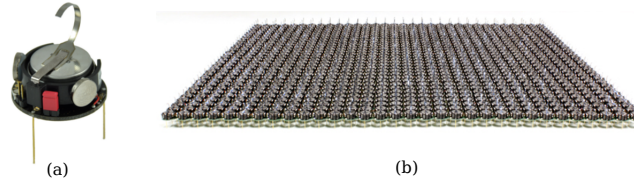


**Figura 7:** Imágenes las partes de un swarmanoid (tomado de [23]) (a) Eye-bot (b) Hand-bots (c) Foot-bot.

## Kilobots

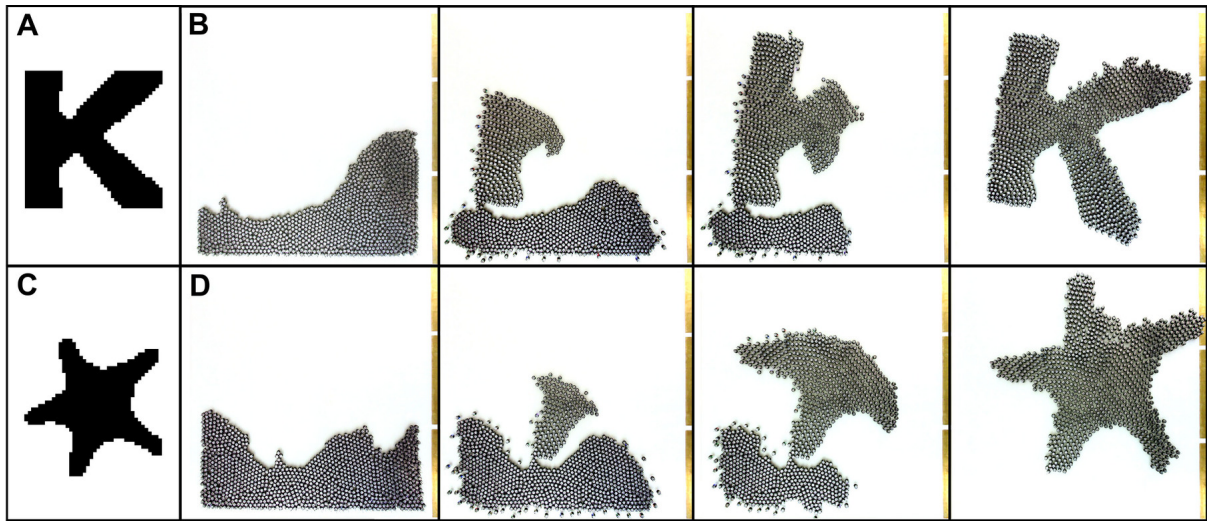
Kilobots es un proyecto desarrollado por la Universidad de Harvard, y consiste en crear un enjambre de mil robots (Ver Figura 8) y diseñar un algoritmo que permita

experimentar con comportamientos colectivos en enjambres autónomos de gran escala. Se demostró que un enjambre de mil robots es capaz de realizar actividades de auto-ensamblaje de manera flexible para crear formas bidimensionales, a través de interacciones y sensores locales [22].



**Figura 8:** (tomado de [22]) (a) Ejemplo de un kilobot individual (b) Enjambre de 1000 kilobots.

Así el algoritmo de auto-ensamblaje une una serie de comportamientos en conjunto con un autómata de estado finitos para lograr crear cualquier forma bidimensional que el usuario desee (Ver Figura 9) [22].



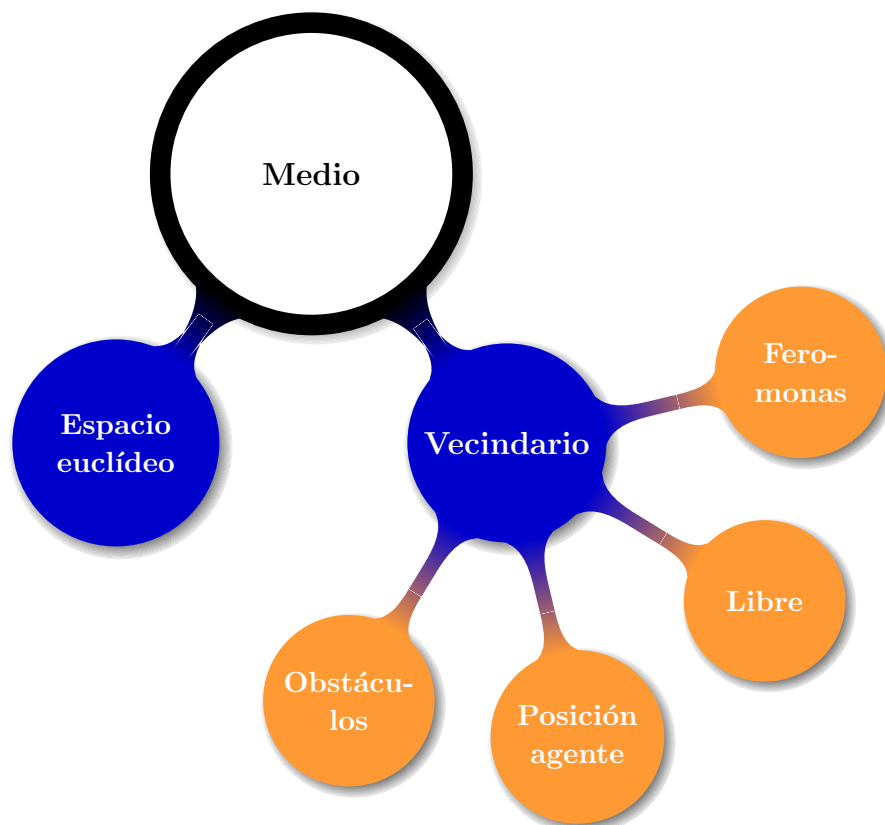
**Figura 9:** Algoritmo de auto-organización de un enjambre de kilobots para lograr crear formas bidimensionales (tomado de [22]). (A, C) Forma bidimensional deseada (B, D) Ejecución del algoritmo.

### 3. Entorno de trabajo del sistema de enjambres

Analizando en conjunto las diferentes aplicaciones y algoritmos existentes sobre enjambres de robots y planificación de rutas en entornos desconocidos descritos en la sección anterior, se describe un algoritmo cooperativo que permite a un grupo de robots autónomos la eficiente exploración de una zona y la posterior comunicación de una ruta hacia un objetivo. En esta sección analizaremos cada una de las partes y definiciones que componen dicho algoritmo. Se trabajó con un sistema de enjambres que consta de dos partes fundamentales: el medio y los agentes.

#### 3.1. Medio

Este elemento tiene su dinámica propia e independiente de lo que sucede con los agentes. Se divide en dos secciones (ver Figura 10: el entorno o espacio el cual indica la zona en la que se encuentran los agentes, y el vecindario que representa la parte del entorno que se encuentra alrededor del agente.



**Figura 10:** Composición del medio de trabajo del sistema de enjambres

A continuación analizaremos de manera más específica cada una de estas partes.

### 3.1.1. Entorno o espacio

Siguiendo el enfoque dado por Marcel et al. en [18], definimos el entorno de la zona a explorar como "un espacio continuo en dos dimensiones que puede tener cualquier número de obstáculos y al menos un punto de entrada y salida". Además, siguiendo un modelo de autómata celular, este entorno corresponde a un espacio Euclídeo finito de  $n$ -dimensiones [25] y se denotará como  $I \times I$

### 3.1.2. Vecindario

Siguiendo el esquema de autómatas celulares, utilizaremos un vecindario de Von Neumann. Así, y siguiendo lo expuesto por Codd en [27], definimos nuestro vecindario (Ver Figura 11) como función  $g : I \times I \rightarrow 2^{I \times I}$  (con  $I \subset \mathbb{Z}$ ) tal que:

$$g(\alpha) = \{\alpha + \delta_1, \alpha + \delta_2, \alpha + \delta_3, \alpha + \delta_4, \alpha + \delta_5\}, \quad \forall \alpha \in I \times I$$

	$\alpha + \delta_2$	
$\alpha + \delta_5$	$\alpha + \delta_1$	$\alpha + \delta_3$
	$\alpha + \delta_4$	

**Figura 11:** Definición del vecindario del espacio celular en el que se trabajará

Los estados posibles para una celda  $\alpha$  en el tiempo  $t$  (los cuales se definirán por medio de números enteros) tenemos:

- **Obstáculo:** consiste en una celda que no le permite a un agente pasar o avanzar hacia un lugar a través de ella. Este estado es inalterable durante todo el tiempo, osea una vez que una celda sea definida como obstáculo lo seguirá siendo durante todo el proceso. Este estado estará representado por:

$$v^t(\alpha) = 0 \quad \forall t$$

- **Posición agente  $X$ :** este estado ocurre cuando alguno de los agentes que conforman el enjambre se encuentra ubicado en una determinada celda. Este estado es variable durante del tiempo y estará representado por:

$$v^t(\alpha) = 0 \quad \text{para algún } t$$

- **Libre:** representa el estado de una celda libre que no tiene presencia de rastro en ella, este estado puede variar en el tiempo. Está representada por:

$$v^t(\alpha) = 2 \quad \text{para algún } t$$

- **Feromona repelente:** este estado suscita luego que un agente haya estado y dejado una celda, dejando atrás un rastro temporal que permite la comunicación del agente con el medio en que se encuentra y viceversa. Dicho rastro se dispersa con el paso del tiempo  $t$ , en general una celda que presente algún tipo rastro de este tipo estará representada por:

$$v^t(\alpha) = -1 \quad \text{para algún } t$$

Cabe destacar que una vez que el rastro se difumine por completo el estado de la celda pasará a ser *Libre* o bien se convertirá inmediatamente en *Posición agente  $X$*  en caso de ser ocupada por un agente.

- **Feromona atrayente:** este estado ocurre luego que un agente que haya encontrado el objetivo o un camino a este, permitiendo a los agentes determinar una ruta hacia el objetivo. Dicho rastro es invariante con el paso del tiempo y estará representada por:

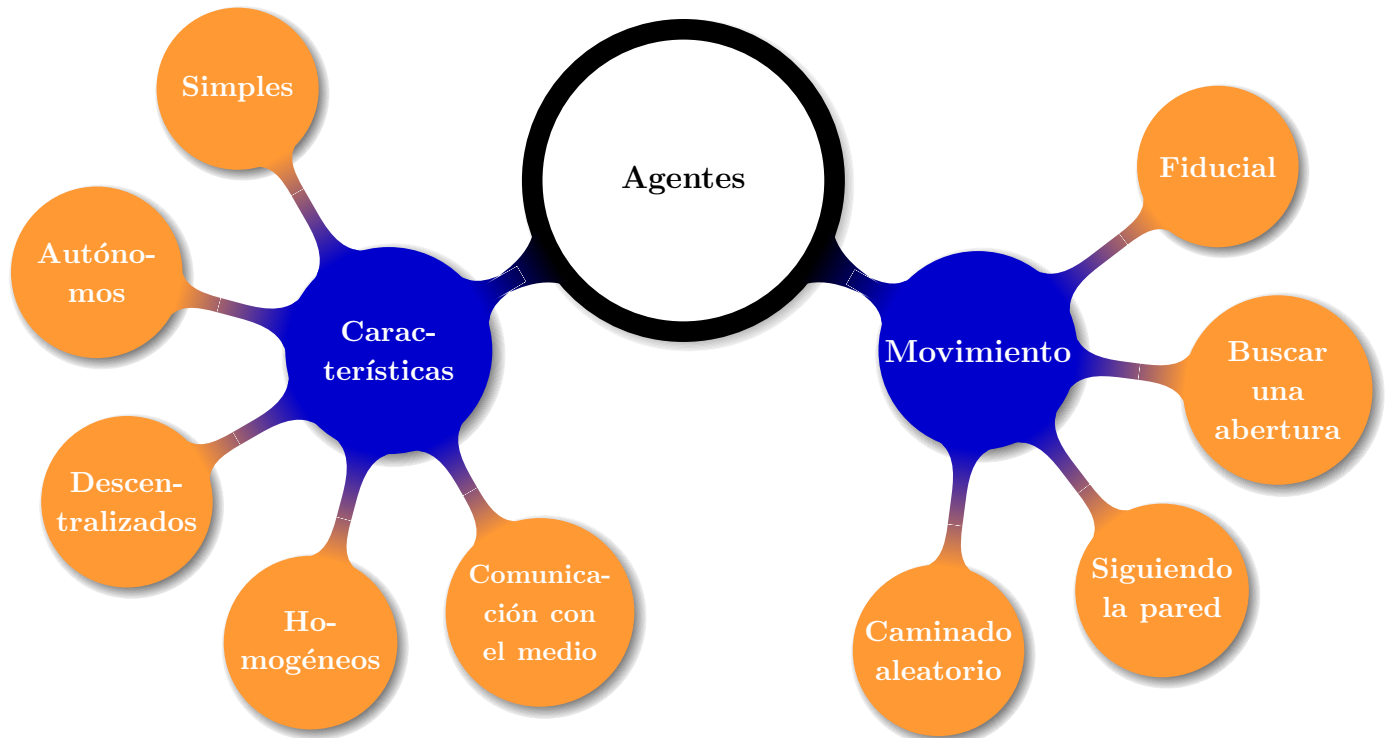
$$v^t(\alpha) = 1 \quad \text{para algún } t$$

- **Meta:** representa el estado de una celda objetivo, o sea representa el lugar al que los agentes deben llegar. Está representada por:

$$v^t(\alpha) = 3 \quad \text{para algún } t$$

### 3.2. Agentes

Los agentes son quienes interactúan directamente con el medio con el fin de obtener información de este y a su vez dejar información relevante sobre el mismo, con el fin de poder comunicarse con otros agentes para propiciar la aparición de un comportamiento colectivo. Están compuestos por dos partes importantes: sus características propias y su movimiento por el medio (ver Figura 12)



**Figura 12:** Composición de los agentes del sistema de enjambres

#### 3.2.1. Características de los agentes

Después de un análisis detallado de las características y la taxonomía de un enjambre, enunciadas en el capítulo anterior y basadas en [2], [3],[4]. Se determinó que los agentes que simulen los robots del enjambre serán:

- **Simples:** tanto en la implementación computacional como en la física se utilizarán los componentes y funciones mínimas para su funcionamiento.

- **Autónomos:** cada agente es independiente de los otros que componen el enjambre. De esta manera el fallo de algún individuo no influye ni pone en riesgo el funcionamiento del algoritmo. Además toman decisiones conforme a la información que encuentren en el ambiente sin que un ente o sistema externo influyan en él.
- **Sistemas descentralizado:** de esta manera los robots funcionan con total independencia sin que un ente o sistema externo influya en su funcionamiento.
- **Comunicación a través del medio ambiente por medio de sensores:** se eligió el tipo de comunicación a través del medio ambiente debido a que representa la forma más simple de comunicación y sigue una inspiración biológica pura. Para lograr este tipo de comunicación se hará uso de sensores para poder interactuar de mejor manera con el entorno.
- **Inspiración biológica:** Se sigue el principio de estigmergia (stigmergy) [29], mediante el uso de feromonas repelentes y atrayentes para comunicarse con el medio y con otros robots. Este comportamiento está basado en la comunicación que realizan sociedades de insectos como hormigas, termitas y otros.
- **Tamaño del enjambre:** se realizarán varias pruebas para determinar la cantidad óptima de robots de acuerdo al entorno que se simulará. Se probará con los tipos *individual, par y grupo limitado* [1].
- **Rango de comunicación:** dado que la comunicación se realizará directamente con el ambiente y no con los otros robots del enjambre se tendrá un rango de comunicación nulo según lo indica en [1].
- **Reconfigurabilidad del enjambre:** dado que se trabaja en un entorno fijo la reconfigurabilidad del enjambre es estático. Sin embargo el sistema está confeccionado para que con pequeños cambios se pueda trabajar con una reconfigurabilidad coordinada (no se trabajó con este tipo dado que se salía de los objetivos propuestos).



- **Composición del enjambre:** los agentes serán homogéneos con el fin de tener un sistema más robusto y tolerable a fallos.

Cabe destacar que los agentes utilizados en esta propuesta no tienen conocimiento previo del medio ambiente en que se desenvuelven, por esto una adecuada comunicación con el medio hará posible la convergencia de los algoritmos que se propongan.

### 3.2.2. Algoritmo de movimiento

En cuanto al movimiento que seguirán los robots, se analizó lo expuesto por Morlok y Gini en [20], en donde se analizaron diferentes algoritmos de movimiento con el fin de determinar aquel que genere dispersar los agentes de manera más eficiente. Entre los algoritmos que se analizaron se encuentran:

1. **Caminado aleatorio (Random Walk):** este es el más básico de los algoritmos de movimiento y consiste en definir en el robot uno de dos estados:
  - *Caminado aleatorio hacia adelante:* en este estado el agente avanza hacia delante al azar, con un pequeño factor de giro aleatorio de entre  $-10^\circ$  y  $10^\circ$ , este factor está constantemente en cambio en intervalos aleatorios que oscila entre 10 seg. y 15 seg. Además la cantidad y ángulos de rotación está restringida para evitar que el robot termine caminando en círculos.
  - *Evitando obstáculos:* este estado ocurre cuando el agente detecta que se ha encontrado con un obstáculo, por lo que él se detendrá y dará un giro aleatorio (en el rango de  $120^\circ$  y  $240^\circ$ ) hasta que logre alejarse del obstáculo. Una vez que haya finalizado este estado de forma exitosa, se procederá a volver al estado de caminado aleatorio hacia adelante.
2. **Siguiendo la pared (Follow the Wall):** este algoritmo se basa en el hecho de que en muchos ambientes como edificaciones, la salida se encuentra contigua a una pared. Por lo que si un robot logra encontrar algún muro y lo sigue, podrá

abarcarse más espacio y encontrar eventualmente la ruta de escape. En este algoritmo, el agente tendrá cuatro estados bien definidos:

- *Encontrar la pared*
- *Alinearse a la pared*
- *Seguir la pared*
- *Manejar las esquinas*

Cabe destacar que si el robot considera que ha perdido la pared, se restablecerá de nuevo al estado original para buscar un nuevo muro para seguir. El principal problema de este algoritmo de movimiento es que asume que cada obstáculo encontrado es una pared, en lugar de tratar de determinar si la entidad observada es algo más pequeño, como un robot a corta distancia. Debido a esto, cuando muchos robots utilizan este algoritmo al mismo tiempo, se tienden a percibir entre sí como paredes y tratan de alinearse unos con otros, lo cual representa un desperdicio de tiempo y recursos.

3. **Buscar una abertura (Seek Open):** este algoritmo propicia que un robot se mueva hacia áreas abiertas en el medio ambiente en que se encuentra. Se basa en el supuesto de que todos los agentes comienzan en la misma área agrupados bastante próximos entre sí. Este algoritmo se ejecuta de la siguiente manera:

- Primero se calcula el vector promedio de obstáculos para todos los obstáculos en el rango del sensor. Este vector se calcula sumando los vectores que apuntan a todos los objetos dentro del rango y luego divide ese número por la cantidad de vectores. De esta manera, la magnitud del vector debe ser grande para objetos cercanos del robot y pequeños para los objetos lejanos.
- Luego el objetivo del robot es moverse en la dirección opuesta del vector promedio obstáculos, la velocidad de giro se determina por la magnitud del vector.

4. **Fiducial:** este algoritmo se inspiró en la idea de que los robots serían capaces de reconocer otros robots, y por lo tanto alejarse de ellos. Para mejorar dicho algoritmo se recomienda dotar a cada robot de un dispositivo que permita conocer las coordenadas polares de los otros robots dentro de su rango. Así el algoritmo general consiste en que cada vez que un robot detecta otro dentro del alcance de su sensor, el robot ajusta su movimiento de modo que se aleje del robot detectado y cuando no hay robots en su rango el agente se mueve de acuerdo con el algoritmo de caminado aleatorio. Si en cualquier momento un robot encuentra un obstáculo físico, como una pared, la técnica de evitación de obstáculos tiene prioridad sobre cualquier algoritmo de movimiento que el robot esté ejecutando.

Dentro de la investigación realizada por Morlok y Gini [20], se demostró que el algoritmo Fiducial es el mejor de todos, pues es el que tiene acceso a más datos, en comparación de los otros algoritmos. Sin embargo se observó que el algoritmo de caminado aleatorio fue el segundo mejor de todos y por lo general, presenta un rendimiento muy cercano al del algoritmo de movimiento Fiducial.

Tomando como base estos resultados, y siguiendo la idea de lograr un algoritmo sencillo, se decidió seleccionar el algoritmo de caminado aleatorio, para ser utilizado por los robots del enjambre. Dado que se estará trabajando en un ambiente celular con un vecindario de Von Neumann, se realizaron algunas modificaciones al algoritmo de movimiento base, el cual se explica a continuación.

Basándonos en la simbología de las funciones de transición de un espacio de 29 estados en un vecindario de Von Neumann, mencionadas por Burks en [25], definimos los siguientes estados de movimiento (Ver Figura 13) :

- *Fijo:* se denotará con el símbolo  $\cup$  y constituye al estado de transición donde un agente se encuentra en una determinada celda (o sea  $v^t(\alpha) = 0$  para algún  $t$ ) y debe determinar a cual otra debe proseguir.
- *Elementos ordinarios de transición:* constituye a las cuatro posibles direcciones que puede tomar un agente que se encuentre en una determinada celda. Estas direcciones corresponden a norte ( $\uparrow$ ), sur ( $\downarrow$ ), este ( $\rightarrow$ ) y oeste ( $\leftarrow$ ).

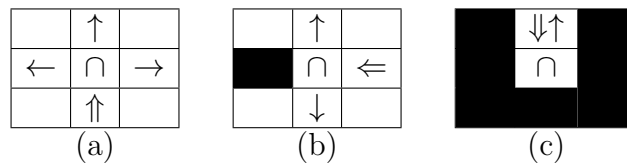
- *Elementos especiales de transición:* indica la dirección previa del agente en el tiempo  $t - 1$ . Estas direcciones son similares a los elementos ordinarios de transición, y corresponden a norte ( $\Uparrow$ ), sur ( $\Downarrow$ ), este ( $\Rightarrow$ ) y oeste ( $\Leftarrow$ ).
- *Nulo:* se denotará con el símbolo  $\cap$  y constituye al estado de transición donde el agente ha encontrado el objetivo por lo que ya no realizará movimiento.

Fijo	{	$\cup$
Elementos ordinarios	{	$\uparrow$ $\downarrow$ $\rightarrow$ $\leftarrow$
Elementos especiales	{	$\Uparrow$ $\Downarrow$ $\Rightarrow$ $\Leftarrow$
Nulo	{	$\cap$

**Figura 13:** Simbología de las funciones de transición de movimiento de un agente

Cabe destacar que un agente no tomará, en el tiempo  $t + 1$ , una dirección (elemento ordinario de transición) contraria a la dirección previa del agente (elemento especial de transición), siempre y cuando exista alguna otra dirección como alternativa.

Dada esta simbología se pueden expresar todos los posibles estados de transición subsecuentes. En la figura 14 se ejemplifican algunos de ellos, (las celdas negras representan celdas con estado obstáculo).



**Figura 14:** Ejemplos de posibles estados de transición de movimiento (a) Agente con dirección anterior Norte (b) Agente con dirección anterior Oeste y un obstáculo (c) Agente con dirección anterior sur y varios obstáculos

### 3.3. Ambiente computacional para la simulación del algoritmo

Para realizar las pruebas de los algoritmos que se propongan, se hará uso de un entorno de desarrollo de procesamiento (PDE, de sus siglas en inglés), en específico el programa Processing. Este programa fue creado por Ben Fry y Casey Reas en el 2001 y en el 2012 se unió Dan Shiffman. Processing fue uno de los primeros proyectos de código abierto que fueron creados especialmente para simplificar la creación de aplicaciones gráficas interactivas, siendo la creación de prototipos de software y visualización de datos, dos de sus áreas más importantes [32].

Processing incluye un entorno de desarrollo integrado (IDE) que se incluye un ambiente para desarrollar las aplicaciones, un lenguaje de programación diseñado específicamente para simplificar la programación para el diseño visual, y las herramientas para publicar las aplicaciones. En general este programa tiene diferentes modos de programación como Java, Android y Python [32].

En este trabajo se utilizó Processing con los siguientes ajustes:

- **Versión:** se utilizó la versión 3.0.1.
- **Modo de programación Java:** este modo permite aprovechar las características propias del lenguaje en conjunto con la simplicidad para realizar aplicaciones gráficas dinámicas.
- **Cuadros por segundo:** para un mejor fluido del movimiento en la animación se utilizaron 50 cuadros por segundo.

En cuanto al entorno de trabajo se modeló por medio de gráficos en dos dimensiones. En general se crearon tres clases para poder simular de manera adecuada el ambiente y los robots, estas clases fueron:

1. **Bot:** es la clase que define los agentes que simularan los robots del enjambre, por lo que contiene todas las funcionalidades asociadas a éstos. Entre los atributos de cada objeto se tienen:

- **Posición:** consiste en un vector que contiene las coordenadas  $x$  y  $y$  del agente dentro del espacio de trabajo.
- **Dirección:** corresponde a un vector que indica la dirección actual del agente. Dado que se trabaja con un vecindario de Von Neumann, sólo existen cuatro posibles direcciones:
  - a) *Norte:* representado por el vector  $(0, -1)$
  - b) *Este:* representado por el vector  $(1, 0)$
  - c) *Sur:* representado por el vector  $(0, 1)$
  - d) *Oeste:* representado por el vector  $(-1, 0)$
- **Aceleración:** es un vector que indica la velocidad de cada agente. En este caso todos los agentes tendrán una velocidad constante.
- **Tamaño:** consiste en un valor entero que define el tamaño del agente.
- **Modo:** corresponde a un número entero que indica el estado del robot (repeliendo feromona, comunicando camino o siguiendo camino al objetivo).

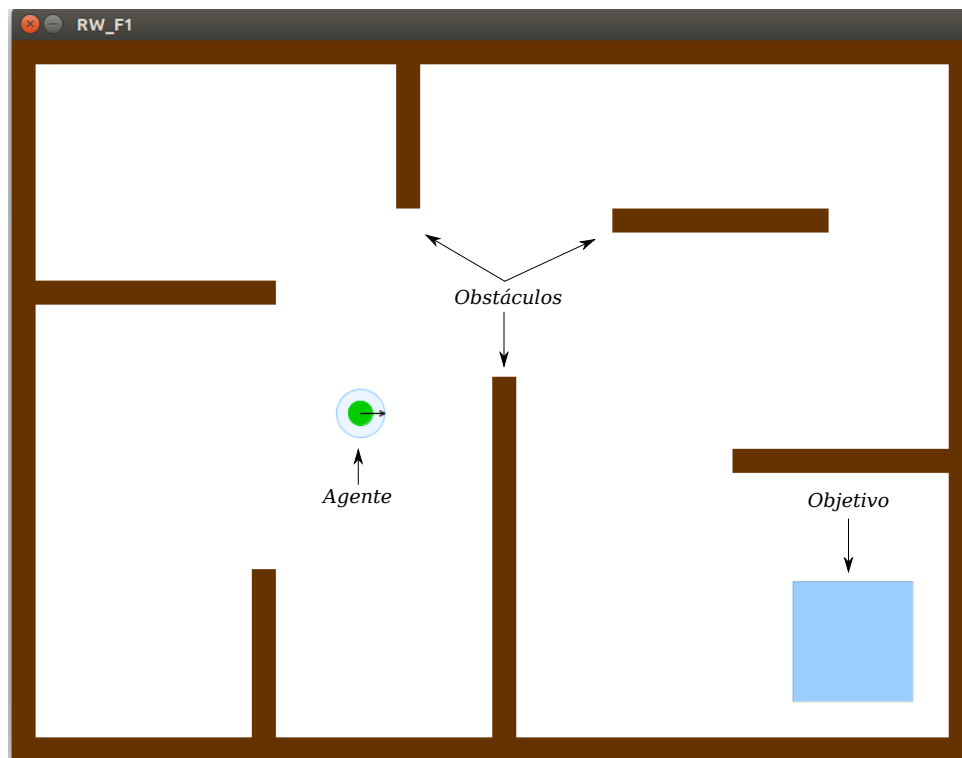
Además de estos atributos cada agente almacena la dirección anterior con la que viajaban y la celda en la que se encontraban en la iteración anterior. Adicionalmente esta clase consta de solamente dos funciones, las cuales corresponde a:

- **Mover:** es la encargada de la animación del movimiento del agente mientras pasa de una celda a otra.
- **Detener:** es la encargada de detener la animación del movimiento del agente debido a que ha llegado al objetivo.

Gráficamente los agentes se componen de tres elementos (Ver Figura 15):

- Un círculo verde que indica el espacio físico del agente.

- Un círculo azul que indica el rango de visión o del sensor del agente.
- Una flecha que indica la dirección que lleva el agente.



**Figura 15:** Entorno computacional para las simulaciones.

2. **EspacioCelular:** corresponde a la clase que se encarga de dividir el espacio de trabajo en celdas, para poder trabajar con un modelo de autómatas celulares. Esta clase tiene los siguientes atributos:
  - **Filas:** es un número entero que indica la cantidad de filas en las que se dividirá el espacio de trabajo.
  - **Columnas:** es similar a la anterior, sólo que indica la cantidad de columnas en las que se dividirá el espacio de trabajo.
3. **Celdas:** corresponde a la clase que se encarga de administrar cada celda del espacio celular. Esta clase tiene los siguientes atributos:

- **Posición:** es par ordenado que indica la posición en que debe ir cada celda del espacio celular.
- **Tamaño:** corresponde a un número que define el tamaño de las celdas.
- **Tipo:** corresponde a una variable entera, la cual indica si una determinada celda corresponde a una posición libre (valor 0) un obstáculo (valor 1) o a una parte del objetivo (valor 3). Gráficamente las celdas “libres” se muestran como espacios en blanco, los obstáculos se muestran de color café y las objetivo se muestran de color celeste (Ver Figura 15).

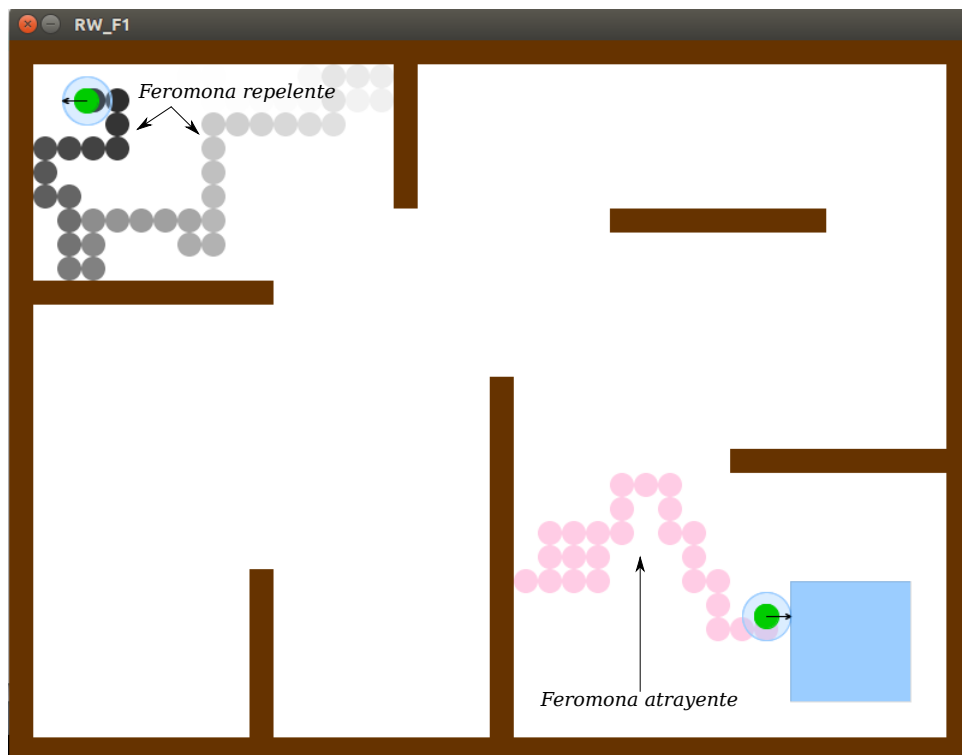
Entre las funciones asociadas a esta clase se encuentran:

- **ActualizarFeromona:** es la encargada de renovar la intensidad de la feromona presente en cada celda, siguiendo el esquema definido en las secciones 4.2 y 5.1.
- **DibujaFeromona:** es la función encargada de mostrar gráficamente el tipo de feromona presente en cada celda y la intensidad de la misma (Ver Figura 16).
- **DibujaTipo:** es similar a la función anterior, se encarga de mostrar gráficamente el tipo de celda (libre, obstáculo, meta).

El equipo computacional en el cual se llevaran a cabo las pruebas consta de las siguientes características:

- Procesador Intel®Core™ i5-4210U CPU @ 1.70GHz × 4.
- Sistema operativo Ubuntu 14.04 LTS de 64 bits.
- Tarjeta gráfica Intel®Haswell Mobile .
- Memoria de acceso aleatorio de 6 GB.





**Figura 16:** Representación del tipo de feromonas

## 4. Algoritmo cooperativo con enjambres de robots para la exploración de una zona en búsqueda de un objetivo

Una vez definido el entorno de trabajo, el tipo de agentes y la simbología para representar sus estados y transiciones, se procede a seleccionar el esquema general que se tomará como base para el algoritmo de exploración de una zona.

Para determinar la eficiencia de los métodos se tomará como métrica la duración de los algoritmos. Este se medirá con base a la cantidad de iteraciones que requirió el algoritmo para ejecutarse. Se tomó la decisión de utilizar iteraciones y no directamente el tiempo de ejecución, con el fin de tener datos más exactos y menos subjetivos a la capacidad computacional del equipo en el cual se estaban ejecutando los algoritmos.

### 4.1. Algoritmo CA: Caminado aleatorio para la búsqueda de un objetivo

Para poder analizar la eficiencia del o los algoritmos que se programen, se tomará como referencia o punto de control, el que resulta de aplicar únicamente el algoritmo de movimiento de *caminado aleatorio* (analizado en la sección 3.2.2) hasta encontrar el objetivo.

En la Figura 17 se puede observar el esquema general del modelo propuesto. Además el pseudocódigo del mismo puede ser observado en el algoritmo el 1. Para este último se utilizó la siguiente simbología:

- $c(i, j)$ : representa cada celda del espacio celular. Cada una puede tener asociada un estado (tal y como se expuso en la sección 3.1.2) como:
  - $c(i, j) = 0$  : Celda que contiene un obstáculo (ya sea uno estático propio del ambiente o un agente que se encuentra en dicha celda).

- $c(i, j) = 3$  : Celda que indica la ubicación de un objetivo.
- $a(k)$ : representa cada agente del enjambre.
- $pos^t(a(k))$ : indica la posición  $(i, j)$  de un agente  $a(k)$  en el espacio celular, en un tiempo  $t$ .
- $vecinos(c(i, j))$ : representa una función que retorna un vector con los cinco vecinos de la celda  $c(i, j)$  tal y como se mostró en la figura 11.

---

**Algoritmo 1 CA:** Caminado aleatorio para la búsqueda de un objetivo

---

**Entrada:** Espacio celular:  $C = J \times J$

Enjambre de  $n$  agentes:  $a$

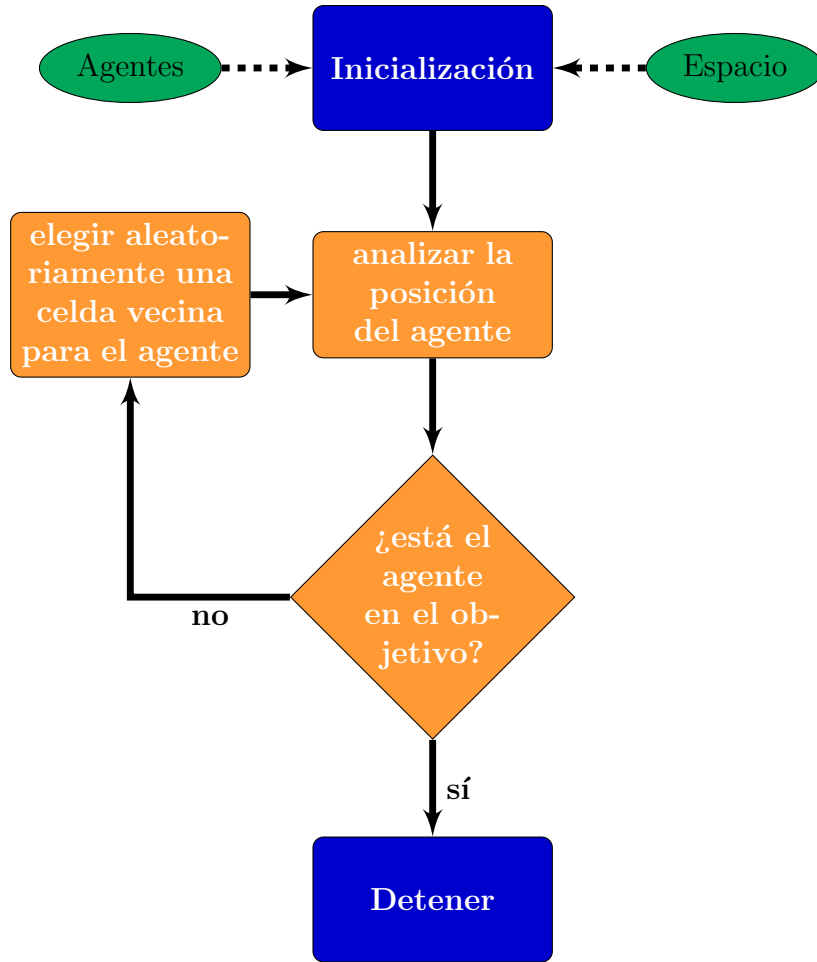
**Salida:**  $id$ : el identificador del agente que encontró el objetivo

```

1:  $id = 0$ 
2: for  $k = 1$  hasta  $n$  do
3:    $(i, j) = pos(a(k))$ 
4:   if  $c(i, j) == 3$  then
5:      $id = k$ 
6:     break
7:   else
8:      $V = vecinos(c(i, j))$ 
9:     if al menos un  $V(l) \neq 0$  then
10:       $m = rand(V(l))$ 
11:      while  $m == 0$  do
12:         $m = rand(V(l))$ 
13:      end while
14:       $pos^{t+1}(a(k)) = m$ 
15:    else
16:       $pos^{t+1}(a(k)) = pos^{t-1}(a(k))$ 
17:    end if
18:  end if
19: end for
20: if  $id \neq 0$  then
21:    $CA(C, a)$ 
22: end if
23: return  $id$ 

```

---



**Figura 17:** Esquema del algoritmo CA: Caminado aleatorio para la búsqueda de un objetivo

#### 4.2. Algoritmo CAFRS: Caminado aleatorio para la búsqueda de un objetivo usando feromonas repelentes

De acuerdo a la literatura revisada y expuesta en la Sección 2, y a los trabajos realizados por diversos autores [18], [19], [28], [30], se tomó la decisión de utilizar un esquema basado en feromonas o “huellas virtuales” repelentes. Esto debido a que dicho enfoque permite que el agente se comunique directamente con el medio ambiente y no con otros agentes de manera directa. Además, las investigaciones citadas anteriormente, probaron la idoneidad de este método para distribuir de manera eficiente a los individuos de un enjambre.

Así el enfoque general que se utilizará consiste en que, conforme los robots van avanzado

por el espacio, irán dejando una marca en cada celda visitada ( $v^t(\alpha) = -1$ ) y asociada a ésta, se tendrá el nivel intensidad de la feromona. Este nivel irá decreciendo en relación con el tiempo transcurrido  $t$  mediante la siguiente función:

$$I(t) = \begin{cases} \frac{I_0}{t - t_0} & \text{si } |t - t_0| \geq I_0 \\ 0 & \text{si } |t - t_0| < I_0 \end{cases}$$

Donde:

- $I_0$  corresponde a la intensidad inicial de la feromona.
- $t_0$  indica el tiempo en que el agente pasó por la celda.

Sin embargo para mayor facilidad se utilizará un modelo iterativo, en el cual la intensidad de la feromona  $I$  presente en una celda  $c$  disminuirá de acuerdo a una tasa de evaporación  $\rho$ , así surge el siguiente modelo:

$$I^{t+1}(c) = \begin{cases} \rho \cdot I^t(c) & \text{si } I^t(c) > 0 \\ 0 & \text{si } I^t(c) \leq 0 \end{cases}$$

Cuando en una celda marcada con feromona repelente se cumpla que la intensidad de la feromona sea 0, esta cambiará su estado a *libre*. En términos técnicos tenemos:

$$(v^t(\alpha) = -1 \wedge P(t) = 0) \Rightarrow v^t(\alpha) = 2$$

Cabe destacar que el valor de  $I_0$  y  $\rho$  debe ser seleccionado de manera que propicie una mejor dispersión de los agentes por el entorno (en la Sección 4.3 se ahondará más sobre este tema).

Luego, conforme los agentes vayan avanzado por el espacio, éstos seguirán el algoritmo de movimiento de caminado aleatorio (analizado en la sección 3.2.2) tomando en consideración la intensidad de la feromona de cada celda vecina para así elegir, aleatoriamente, entre las que tengan menor intensidad hasta llegar a un cierto objetivo.

Con base a estas directrices se procedió a crear el algoritmo 2, el cual utiliza la misma simbología empleada en el algoritmo 1 agregando la función  $orden(V)$ , la cual ordena de manera ascendente los vecinos de una celda de acuerdo a la intensidad de feromona presente en ellas. En esquema del algoritmo puede ser observado en la Figura 18.

---

**Algoritmo 2 CAFRS:** Caminado aleatorio para la búsqueda de un objetivo usando feromonas repelentes

---

**Entrada:** Espacio celular:  $C = J \times J$

Enjambre de  $n$  agentes:  $a$

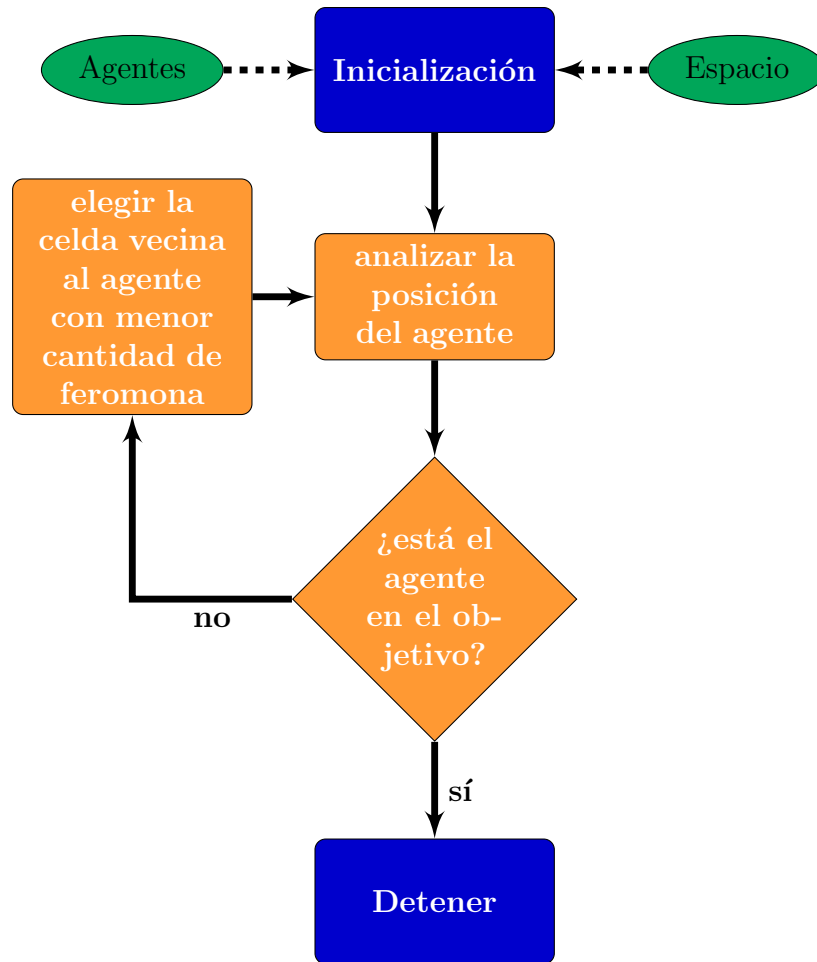
**Salida:**  $id$ : el identificador del agente que encontró el objetivo

```

1:  $id = 0$ 
2: for cada  $c$  en  $J \times J$  do
3:    $I(c) = \rho \cdot I(c)$ 
4: end for
5: for  $k = 1$  hasta  $n$  do
6:    $(i, j) = \text{pos}(a(k))$ 
7:   if  $c(i, j) == 3$  then
8:      $id = k$ 
9:     break
10:  else
11:     $V = \text{vecinos}(c(i, j))$ 
12:    if al menos un  $V(l) \neq 0$  then
13:       $m = \text{orden}(V(l))$ 
14:       $\text{min}V = I(m(1))$ 
15:       $m = \text{rand}(V(l))$ 
16:      while  $I(m) \neq \text{min}V$  do
17:         $m = \text{rand}(V(l))$ 
18:      end while
19:       $\text{pos}^{t+1}(a(k)) = m$ 
20:       $I(m) = I_0$ 
21:    else
22:       $\text{pos}^{t+1}(a(k)) = \text{pos}^{t-1}(a(k))$ 
23:       $I(m) = I_0$ 
24:    end if
25:  end if
26: end for
27: if  $id \neq 0$  then
28:   CAFRS( $C, a$ )
29: end if
30: return  $id$ 

```

---



**Figura 18:** Esquema del algoritmo CAFRS: Caminado aleatorio para la búsqueda de un objetivo usando feromonas repelentes

### 4.3. Pruebas experimentales del algoritmo CAFRS

Una vez confeccionados los algoritmos, se procede a realizar una serie de pruebas con el fin de poder determinar la eficiencia del algoritmo 2 en comparación con el algoritmo 1. Se utilizó el ambiente y equipo computacional indicado en la sección 3.3.

Para analizar de mejor manera los algoritmos, se usaron los siguientes parámetros de comparación:

- **Cantidad de iteraciones:** se refiere al número de “pasos” necesarios por un agente del enjambre para poder llegar al objetivo. Para este apartado se toma en consideración únicamente la cantidad que necesitó el primer agente que encontró

el objetivo.

- **Cantidad de agentes en el enjambre:** para analizar la pertinencia de utilizar múltiples agentes y determinar la cantidad mínima óptima de éstos para resolver el problema de búsqueda de un objetivo, se realizaron pruebas con 1, 3, 5 y 10 agentes.
- **Intensidad de la feromona:** en esta primera parte de las pruebas, se utilizó una intensidad inicial de 500 unidades por iteración, con una tasa de evaporación de  $\rho \approx 0,99$ .
- **Cantidad de repeticiones:** con el fin de obtener datos estadísticos relevantes se realizaron 10 repeticiones de cada uno de los modelos.

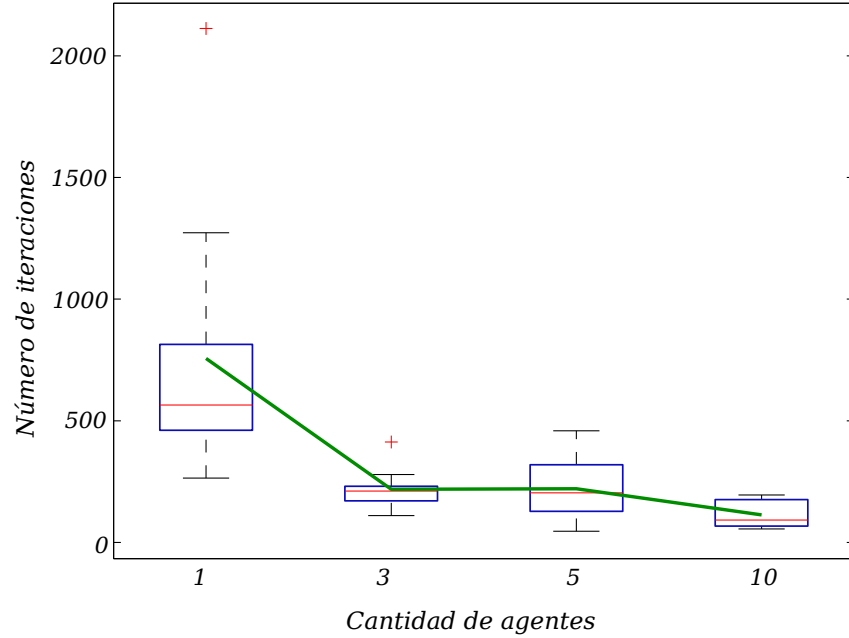
Primero se ejecutó el esquema de caminado aleatorio (Algoritmo 1) utilizando los parámetros definidos anteriormente. Después de ejecutar las 10 repeticiones (los resultados se encuentran desde el anexo A.1 hasta el A.4) y determinar las medidas estadísticas correspondientes (Ver Figura 19) se puede observar que a medida que aumenta la cantidad de agentes en el enjambre, el tiempo requerido por el algoritmo para encontrar el objetivo es cada vez menor.

Además entre más agentes se tengan la dispersión de los datos tiende a ser menor, pues se pasa de una desviación estándar de 558.4 con un robot a una desviación de 56.5 con diez agentes.

Los resultados obtenidos son consistente con la teoría general de enjambres, la cual establece que entre más agentes haya más rápido convergerá un algoritmo. Sin embargo, como los agentes no tienen conocimiento de la zona en la que se desenvuelven, se está propenso a que en alguna de las ejecuciones se obtengan datos atípicos que distan del comportamiento normal del modelo. En este caso, se puede observar que esto ocurrió dos veces, una cuando se trabajó con 1 agente y otra cuando se utilizaron 3 de ellos (Ver Figura 19).

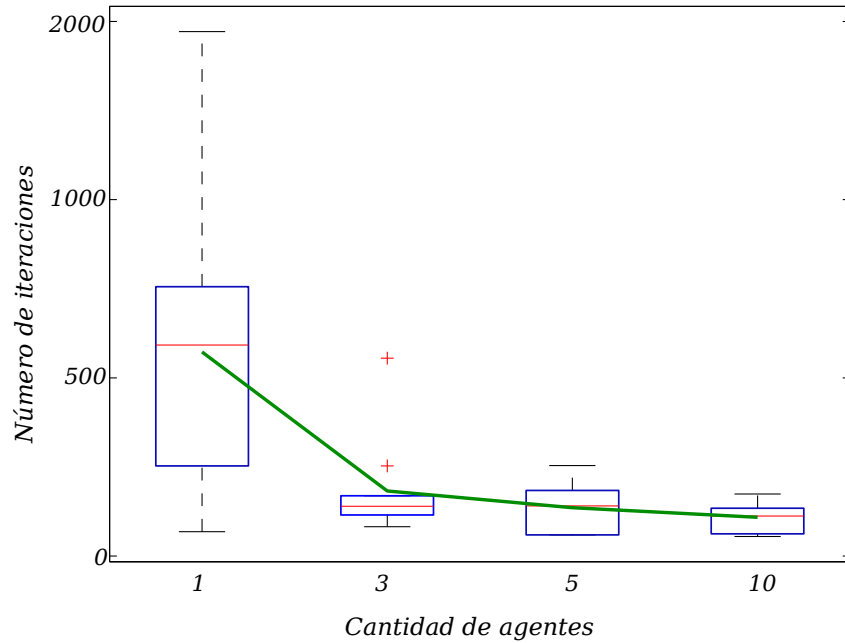


Dados que estos valores ocurrieron sólo una vez en cada uno de estos dos casos y el resto de datos se mantuvieron dentro del rango normal, podemos concluir que dichos valores no afectan de manera significativa las medidas estadísticas calculadas.



**Figura 19:** Resumen estadístico de la ejecución del algoritmo de caminado aleatorio

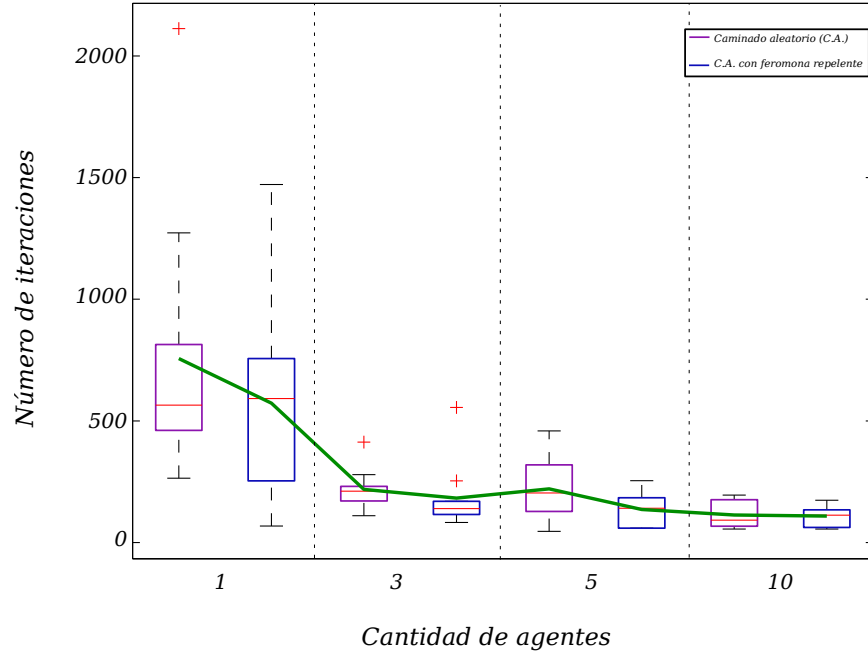
Por otro lado, al ejecutar el esquema de caminado aleatorio con feromonas repelentes (algoritmo 2) y utilizando los parámetros definidos al inicio de la sección (los resultados específicos de cada repetición se encuentran desde el anexo A.5 hasta el A.8). Se observa que, al igual que sucedió con el esquema de caminado aleatorio simple, conforme aumenta la cantidad de agentes en el enjambre el tiempo requerido para encontrar el objetivo es cada vez menor (Ver Figura 20).



**Figura 20:** Resumen estadístico de la ejecución del algoritmo de caminado aleatorio con feromonas repelentes

En general, todos los datos obtenidos se comportaron de manera consistente, por lo cual se dice que las medidas estadísticas obtenidas son relevantes para el análisis de los resultados. Se puede observar que en la ejecución del algoritmo con 3 agentes se obtuvieron dos datos atípicos los cuales alteraron levemente el cálculo del promedio de iteraciones requeridas para localizar el objetivo (Ver Figura 20).

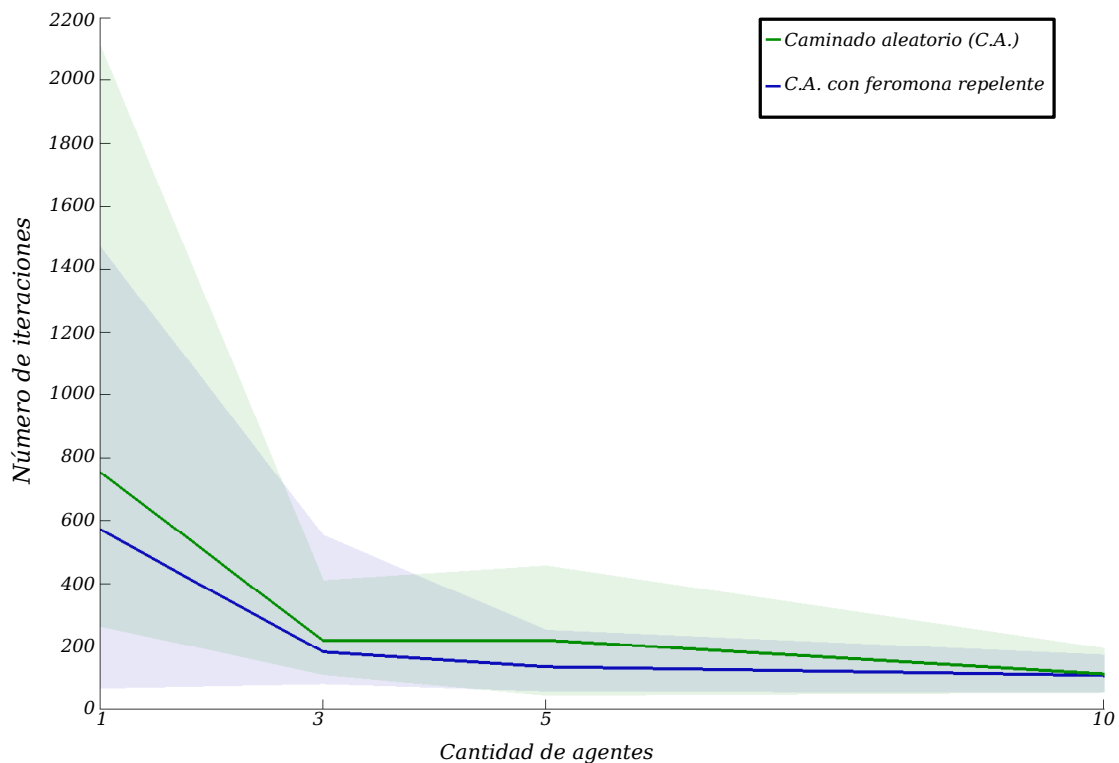
Al analizar en conjunto los datos obtenidos por ambos algoritmos (Ver Figura 21 y Figura 22), se puede observar como el algoritmo de caminado aleatorio con feromonas repelentes presenta un mejor rendimiento que el del caminado aleatorio simple.



**Figura 21:** Comparación de datos estadísticos de los modelos de caminado aleatorio simple y caminado aleatorio con feromonas repelentes

En la mayoría de los experimentos realizados con el esquema de feromonas repelentes, se nota una mejoría en casi todas las medidas estadísticas en cada uno de los casos de análisis, demostrando así la pertinencia y efectividad de dicho algoritmo. Además los resultados obtenidos son consistentes con los encontrados por Fossum en su tesis de maestría “Repellent pheromones for effective swarm robot search in unknown environments”, en donde demuestra que al utilizar un esquema de feromonas repelentes los agentes se distribuyen de manera más efectiva por el medio [19].

También se puede observar que los datos obtenidos por el algoritmo 2 se encuentran más compactos que los del esquema de caminado aleatorio simple. Esto muestra que al utilizar feromonas repelentes, los datos tienden a converger de mejor manera.

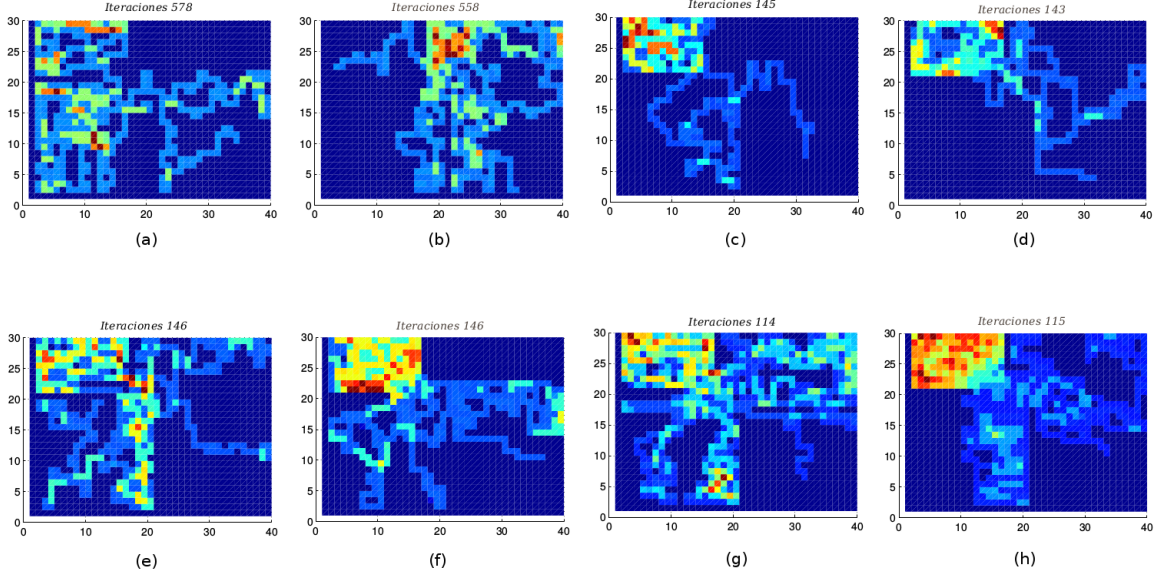


**Figura 22:** Comparación general de los modelos de caminado aleatorio simple y caminado aleatorio con feromonas repelentes

También se puede observar que conforme aumenta la cantidad de agentes en el enjambre, las diferencias entre los dos algoritmos son cada vez menores, esto se debe a que entre más agentes se tengan será menor el área a explorar por cada uno de ellos, produciendo así que se encuentre el objetivo de una manera más rápida.

Por otro lado, si comparamos el recorrido realizado por los agentes en cada uno de los esquemas (Ver Figura 23), podemos observar como los que siguen el modelo de feromonas tienden a repetir menos lugares previamente visitados.

Un detalle importante que arrojan estos resultados es que, debido a que los agentes se liberan en una zona relativamente cerrada (sólo tiene una salida tal y como se mostró en la Figura 16), tienden a re-explorar mucho este sitio debido a que se les dificulta encontrar la salida a este lugar.



**Figura 23:** Recorrido de los agentes. (a) Un agente modelo simple (b) Un agente modelo con feromonas (c) Tres agentes modelo simple (d) Tres agentes modelo con feromonas (e) Cinco agentes modelo simple (f) Cinco agentes modelo con feromonas (g) Diez agentes modelo simple (h) Diez agentes modelo con feromonas. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

#### 4.4. Algoritmo CAFRM: Modificación al algoritmo de caminado aleatorio usando feromonas repelentes

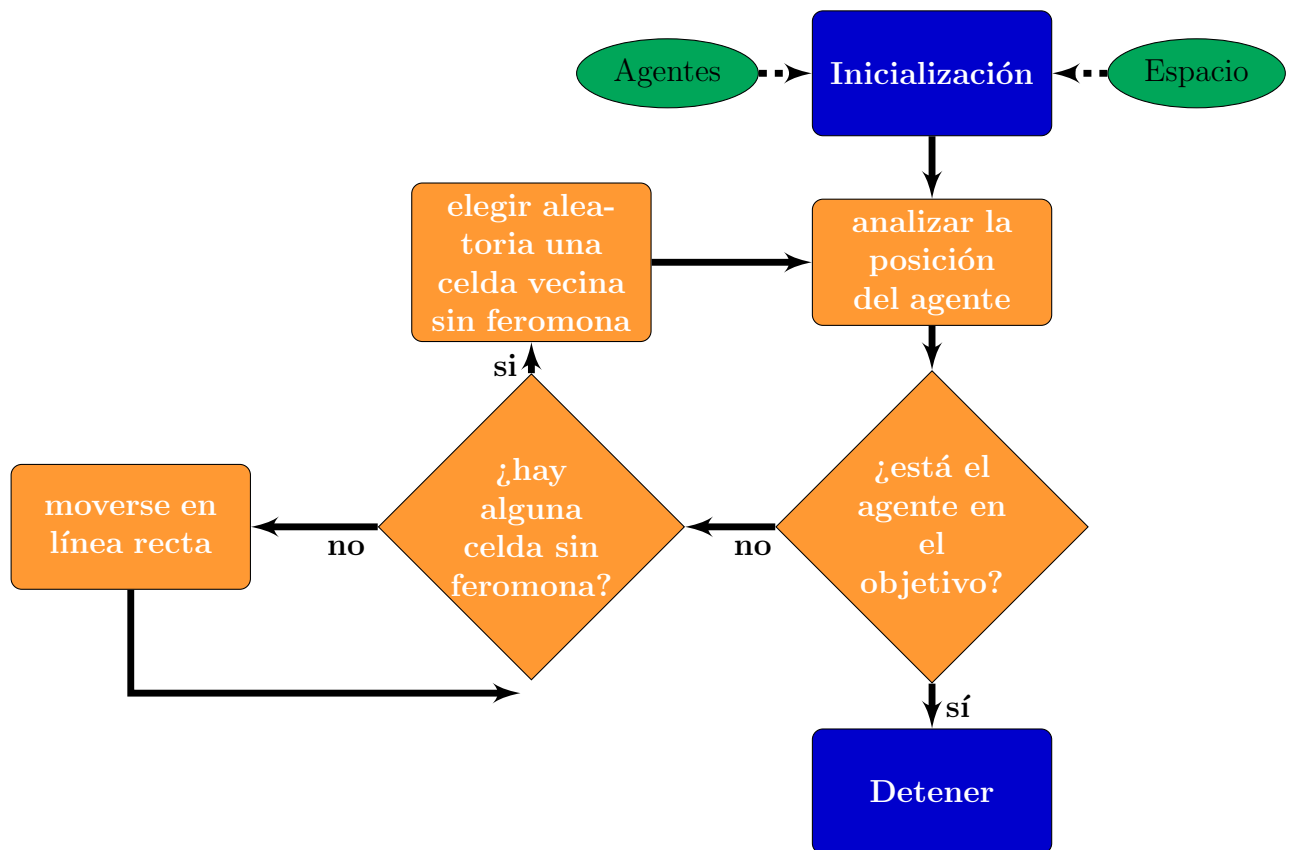
Luego del análisis realizado en la sección anterior, se propone una modificación del algoritmo 2, con el fin de reducir la re-examinación de zonas. La idea general de este nuevo algoritmo es que los agentes detecten si se encuentran en una locación que esta siendo explorada por otros robots y, de ser así, intenten salir lo más pronto posible de este territorio.

Dado que los individuos del enjambre no pueden comunicarse de manera directa entre ellos, y que su rango de visión y memoria es relativamente pequeña, un agente determinará que se encuentra en una zona recientemente visitada si la cantidad de la feromonas en cada una de sus celdas vecinas es mayor a un cierto factor de intensidad  $\tau$ , osea:

$$I(\alpha + \delta_i) > \tau \quad \forall i = 2, 3, 4, 5$$

Una vez que el agente haya determinado que se encuentra en una zona de este tipo, dejará de moverse de acuerdo al algoritmo de caminado aleatorio analizado anteriormente; y optará por una versión modificada de este mismo. Esta nueva versión consiste en moverse en línea recta hacia una determinada dirección (norte, este, sur u oeste) y cuando se encuentre con un obstáculo elegir aleatoriamente una nueva dirección para seguir el movimiento rectilíneo. Así el agente seguirá este algoritmo de movimiento hasta que logre llegar a una zona “despejada”.

Así al unir estas dos ideas se obtiene un nuevo algoritmo 3, el cual se representa de mejor manera en la Figura 24). Cabe destacar que se utilizó la misma simbología de la sección tras anterior, y se agregó la función  $nuevaDir(V, a)$  que retorna la celda a la que debe moverse el agente, tomando en cuenta la dirección actual del robot y las celdas vecinas.



**Figura 24:** Esquema del algoritmo CAFRM: Modificación al algoritmo de caminado aleatorio usando feromonas repelentes

---

**Algoritmo 3 CAFRM:** Modificación al algoritmo de caminado aleatorio usando feromonas repelentes

---

**Entrada:** Espacio celular:  $C = J \times J$

Enjambre de  $n$  agentes:  $a$

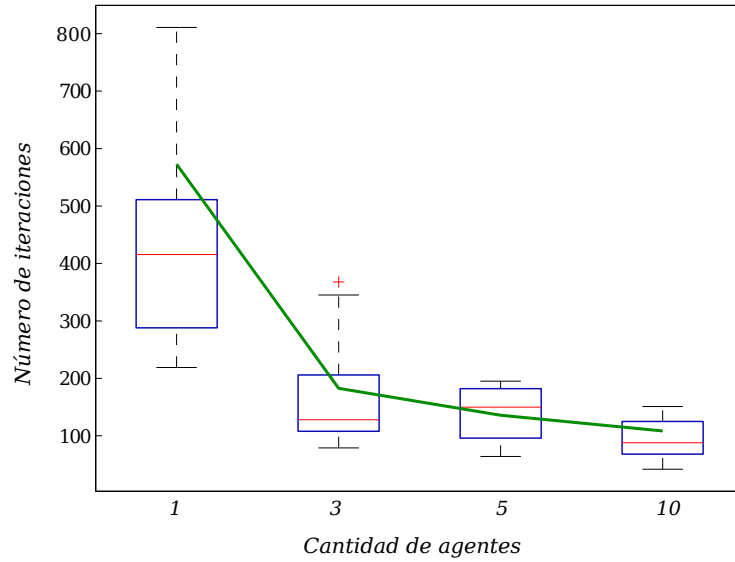
**Salida:**  $id$ : el identificador del agente que encontró el objetivo

```
1:  $id = 0$ 
2: for cada  $c$  en  $J \times J$  do
3:    $I(c) = \rho \cdot I(c)$ 
4: end for
5: for  $k = 1$  hasta  $n$  do
6:    $(i, j) = \text{pos}(a(k))$ 
7:   if  $c(i, j) == 3$  then
8:      $id = k$ 
9:     break
10:  else
11:     $V = \text{vecinos}(c(i, j))$ 
12:    if al menos un  $V(l) \neq 0$  then
13:       $m = \text{orden}(V(l))$ 
14:       $\min V = I(m(1))$ 
15:      if  $\min V > \tau$  then
16:         $m = \text{nuevaDir}(V, a)$ 
17:         $\text{pos}^{t+1}(a(k)) = m$ 
18:      else
19:         $m = \text{rand}(V(l))$ 
20:        while  $I(m) \neq \min V$  do
21:           $m = \text{rand}(V(l))$ 
22:        end while
23:         $\text{pos}^{t+1}(a(k)) = m$ 
24:         $I(m) = I_0$ 
25:      end if
26:    else
27:       $\text{pos}^{t+1}(a(k)) = \text{pos}^{t-1}(a(k))$ 
28:       $I(m) = I_0$ 
29:    end if
30:  end if
31: end for
32: if  $id \neq 0$  then
33:   CAFRM( $C, a$ )
34: end if
35: return  $id$ 
```

---

## 4.5. Pruebas experimentales del algoritmo CAFRM

Con base a este nuevo algoritmo, se procedió a ejecutarlo utilizando los mismos parámetros definidos en la sección 4.3 (los resultados específicos de cada repetición se encuentran desde el anexo A.9 hasta el A.12). Al igual que con los algoritmos 1 y 2, se observa como el tiempo requerido para encontrar un objetivo es menor conforme aumenta el número de agentes (Ver Figura 25).

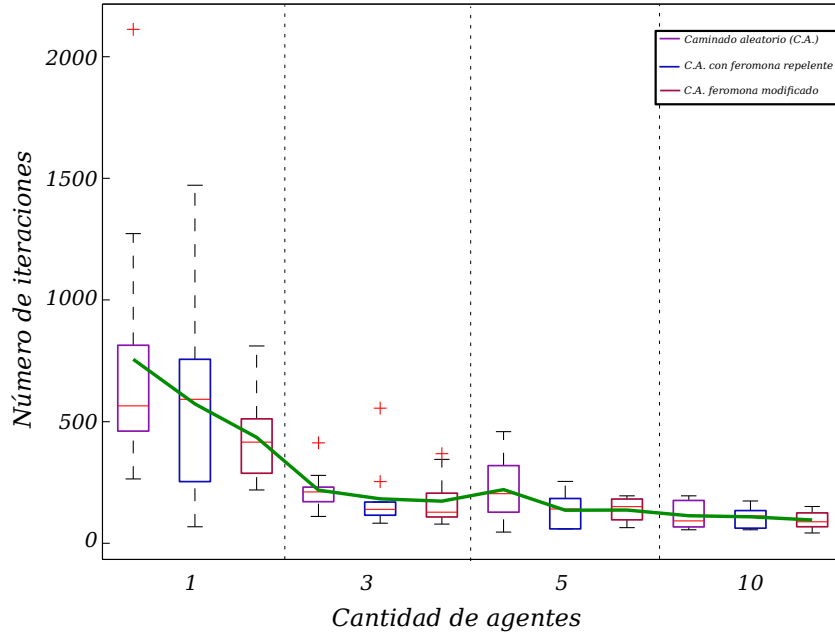


**Figura 25:** Resumen estadístico de la ejecución del algoritmo modificado de caminado aleatorio con feromonas repelentes

En este caso se produjo solamente un dato atípico al utilizar 3 agentes. Se puede observar en la Figura 25, como este dato se encuentra muy cerca del límite superior de la muestra, por lo que se puede afirmar que dicho dato no afecta de manera significativa los datos estadísticos obtenidos.

Al analizar los 3 métodos en conjunto se puede observar como el algoritmo 3 mejora levemente los resultados obtenidos por el algoritmo 2 y en mayor medida los obtenidos con el algoritmo 1 (ver Figura 26).

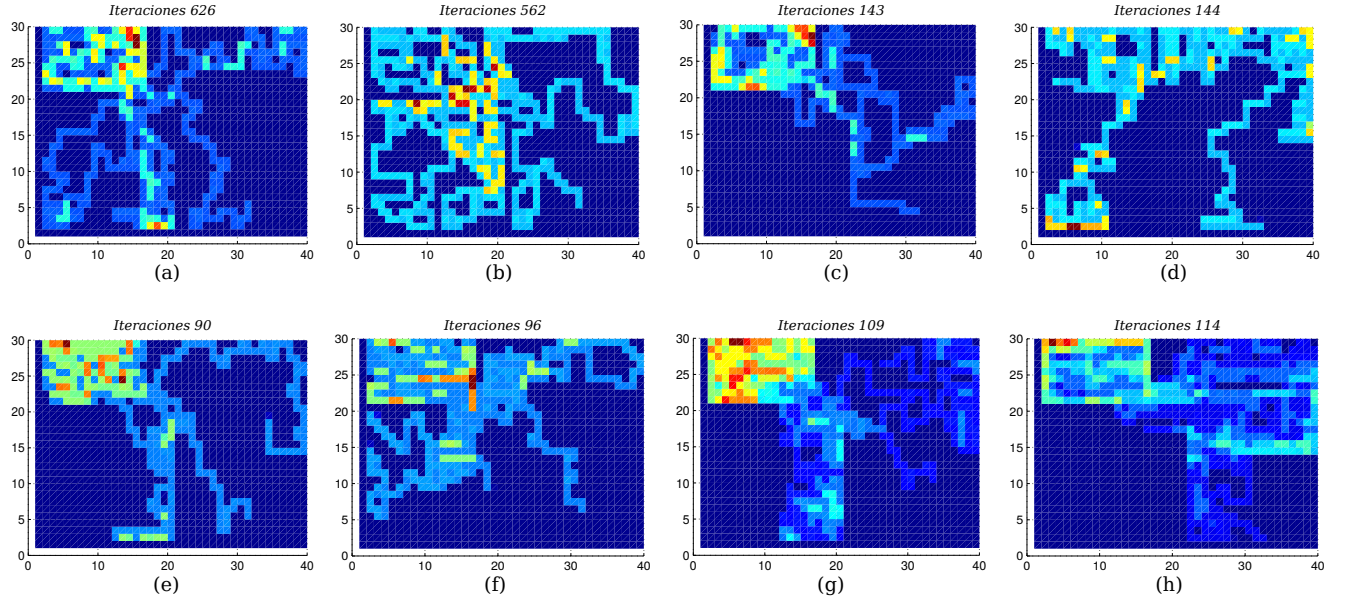




**Figura 26:** Comparación de datos estadísticos de los modelos de caminado aleatorio simple, caminado aleatorio con feromonas repelentes y modificación del caminado aleatorio con feromonas repelentes

Además los datos se encuentran, en casi todos los casos, más compactos. De hecho para el caso de 1 agente se pasó de una desviación estándar de 558.4 con el primer algoritmo a 408.7 con el segundo y finalmente a 169 con este tercer enfoque. Al utilizar 10 agentes la disminución, aunque menor, también se hizo notoria pues se pasó de un 56.5 a un 40.2 y luego a un 37.2 en los tres algoritmos respectivamente.

Otro punto importante es que se produjo una disminución en la re-examinación de zonas, lo que produce una mejor dispersión del enjambre en el ambiente. Esto lo podemos observar en la Figura 27, donde se compara el recorrido realizado por los agentes en los algoritmos CAFRS y CAFRM respectivamente. En estas pruebas se compararon ejecuciones que hayan generado una cantidad similar de iteraciones, con el fin de observar mejor la mejora realizada por el algoritmo 3



**Figura 27:** Recorrido de los agentes en los modelos de caminado aleatorio con feromonas repelentes simple y modificado. (a) Un agente modelo simple (b) Un agente modelo modificado (c) Tres agentes modelo simple (d) Tres agentes modelo modificado (e) Cinco agentes modelo simple (f) Cinco agentes modelo modificado (g) Diez agentes modelo simple (h) Diez agentes modelo modificado

#### 4.6. Análisis de resultados de los algoritmos cooperativos con enjambres de robots para la exploración de una zona en búsqueda de un objetivo

Con base a los resultados obtenidos al ejecutar los tres algoritmos CA, CAFRS y CAFRM se logra evidenciar que:

- Entre más agentes tenga un enjambre más rápido convergerá un algoritmo.
- Cuando se aumenta la cantidad de individuos las diferencias del tiempo promedio de convergencia entre cada método serán cada vez menores.
- Con mayor cantidad de agentes, la dispersión de los datos será cada vez menor.
- El algoritmo CAFRM es el que mejor rendimiento presenta (ver Figura 26), mejorando en todos los casos los resultados obtenidos con el algoritmo base CA y de manera similar con los producidos por el algoritmo CAFRS.

- En todos los algoritmos se presentan datos atípico dentro del rango normal de los resultados, esto sucede por tratarse de un sistema aleatorio donde no se tiene control total sobre las acciones de los agentes. Sin embargo dichos datos son pocos y no muy alejados del rango normal, por lo que no afecta de manera significativa los medidas estadísticas calculadas.

Todas las pruebas realizadas hasta el momento se realizaron empleando una intensidad de feromona inicial de 500 unidades. Por lo que para medir el impacto de de dicho valor, se procede a realizar una serie de pruebas que pretenden analizar la cantidad óptima inicial para este problema en específico.

Para las pruebas se utilizaron 10 repeticiones de los algoritmos 2 y 3 con intensidades de 50, 100, 300, 500, 700, 900 y 1200, con el fin de poder analizar mejor dos aspectos fundamentales:

1. La eficiencia del algortimo de caminado aleatorio con feronomas repelentes modificado vs la versión simple.
2. El impacto de la intensidad de la feromona en la velocidad de la convergencia de cada uno de los algoritmos.

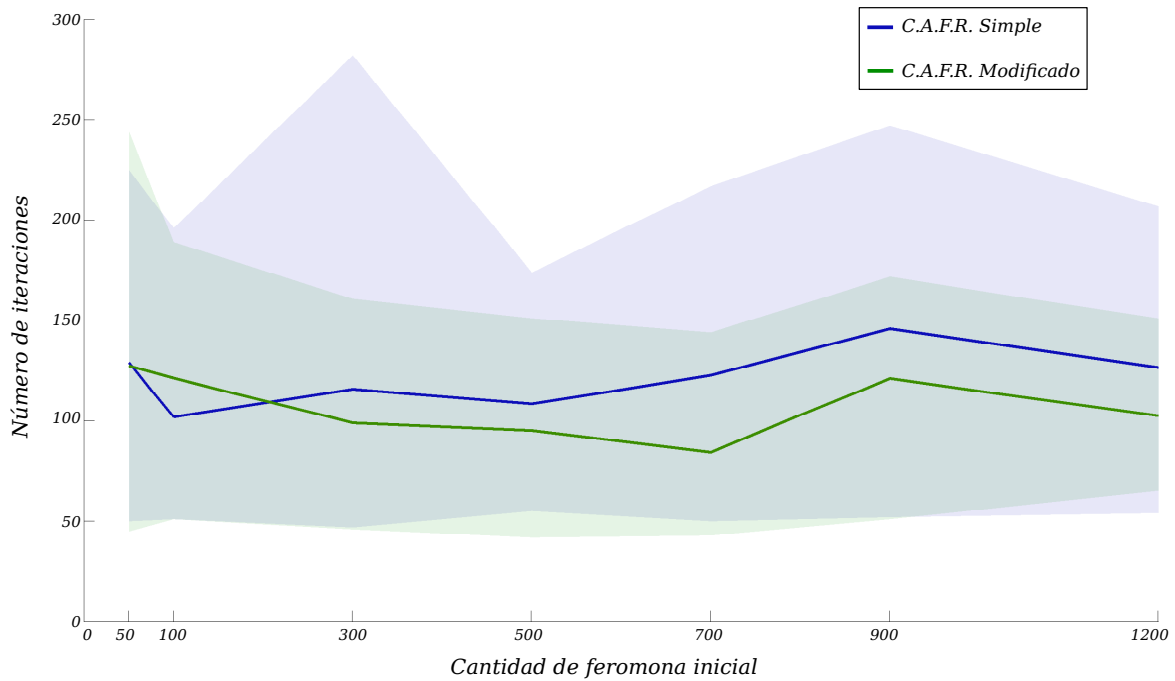
En Figura 28 se pueden observar los resultados obtenidos al ejecutar cada uno de los algoritmos con las intensidades indicadas.

Con el algoritmo simple se obtiene un mejor desempeño cuando se asigna una intensidad inicial de 100, pero conforme se va aumentando dicho valor el tiempo promedio de duración va aumentando ligeramente.

Con relación al algoritmo 3, se tiene que el menor tiempo promedio de duración se obtiene al asignar una intensidad inicial de 700; y conforme se disminuye o aumenta este valor, el tiempo de duración aumenta pero de manera muy leve. Mientras que con el algoritmo CAFRS se obtiene dicho valor óptimo, cuando se asigna una intensidad inicial de 50.

Estos datos obtenidos son válidos para el entorno actual de trabajo. También se puede observar que el algoritmo mejorado converge en casi todos los casos de manera más

rápida que el algoritmo simple (ver Figura 28).



**Figura 28:** Comparación métodos de caminado aleatorio con feromonas repelentes simple y modificado

Con base a todos estos datos y pruebas, se concluyen que el esquema de caminado aleatorio modificado con feromonas repelentes representado en el algoritmo 3 es el que mejor desempeño presenta, por lo que en adelante se trabajará con dicho algoritmo, indicando una intensidad inicial de 700 para las feromonas repelentes.

## **5. Algoritmo cooperativo con enjambres de robots para la comunicación de una ruta hacia un objetivo**

Una vez definido el algoritmo de búsqueda, se propone un esquema general para determinar la mejor manera de comunicar el camino hacia un objetivo.

### **5.1. Algoritmo COFAS: Comunicación de un objetivo usando feromonas atrayentes**

Continuando con la idea de utilizar feromonas para interactuar con el medio y con los individuos del enjambre, se decidió seguir utilizando este esquema para implementar un algoritmo que permita comunicar el camino hacia un objetivo.

En la sección anterior, vimos como se usaron feromonas pero con un enfoque repelente para lograr distribuir de manera más efectiva a los individuos del enjambre por el medio, esto con el fin de poder localizar un objetivo de una rápida. Sin embargo, como la idea de este apartado es encontrar un algoritmo que permita comunicar de manera eficiente y rápida el camino hacia un objetivo, se propone utilizar un esquema de feromonas atrayentes.

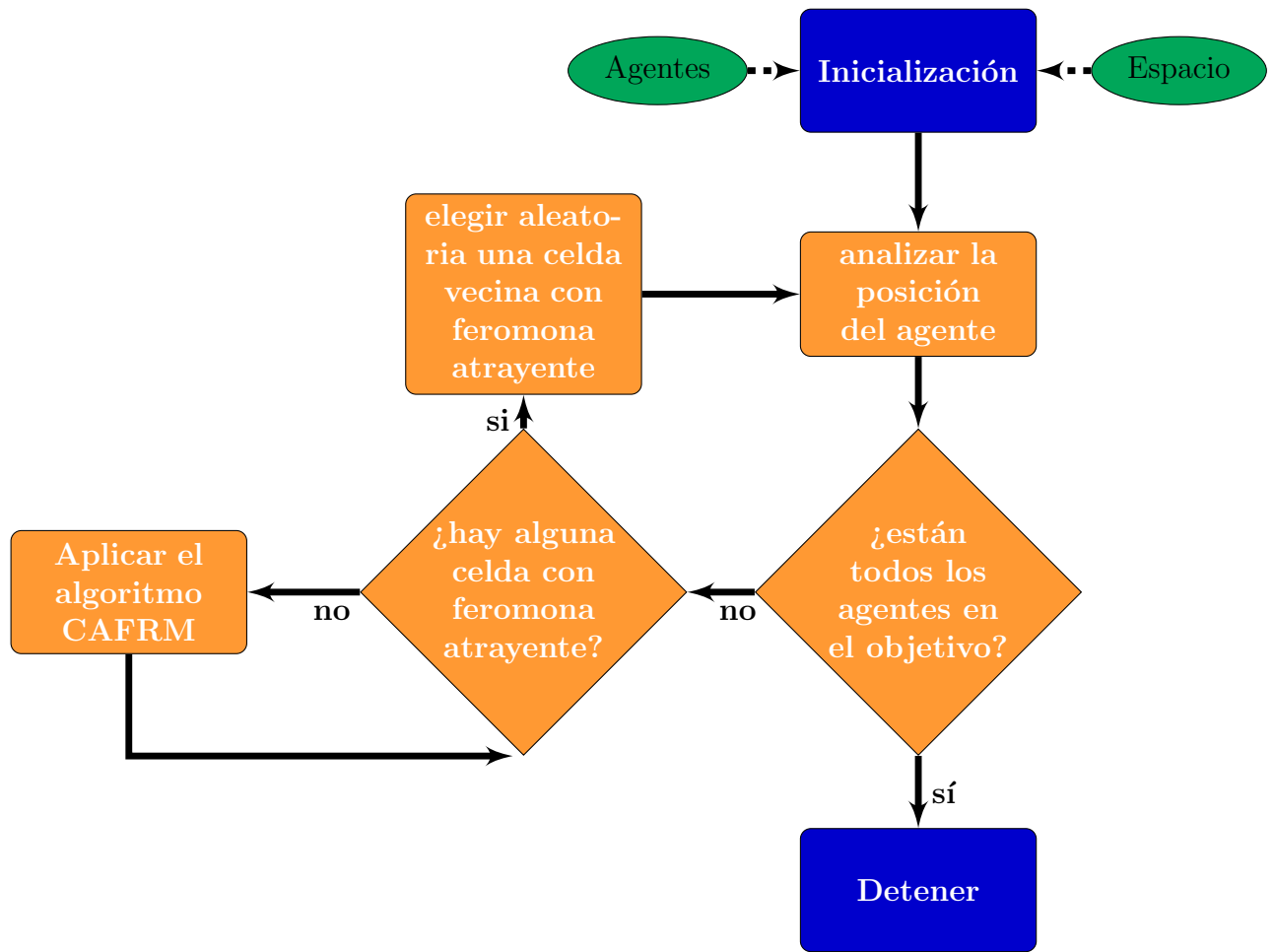
Este esquema es el utilizado por sociedades de insectos como las hormigas y termitas, los cuales dejan un rastro en el ambiente con el fin de poder comunicar el camino hacia una fuente de comida u otros [13], [28], [29].

Así y siguiendo esta idea, se propone que una vez que un agente llegue al objetivo se devuelva una cierta cantidad de pasos dejando en su camino una feromona atrayente, para luego regresar al objetivo. Además, si un individuo llega a un lugar donde sienta dicha feromona, se devolverá una cantidad igual de pasos, para luego seguir el rastro hacia el objetivo.

Además se utilizará una feromona permanente, osea el rastro de la misma no variará con el tiempo, esto con el fin de que el algoritmo converja de manera más rápida. Para

facilitar la búsqueda del camino de regreso a la meta, se decidió que los agentes utilizaran un algoritmo de movimiento aleatorio rectilíneo, una vez que este haya encontrado al objetivo o una ruta hacia él.

Recordemos que este algoritmo de movimiento consiste en moverse en línea recta hacia una determinada dirección (norte, este, sur u oeste) y cuando se encuentre con un obstáculo elegir aleatoriamente una nueva dirección para seguir el movimiento rectilíneo. El esquema general de dicho algoritmo se puede observar en la Figura 29.



**Figura 29:** Esquema del algoritmo COFAS: Comunicación de un objetivo usando feromonas atrayentes

En el algoritmo 4, se puede observar el pseudocódigo del esquema general presentado anteriormente. En este se empleó la misma simbología utilizada en la sección anterior, donde además se agregan las siguientes funciones y definiciones:

- $c(i, j) = -1$  : representa una celda que contiene feromona repelente. Dado que este estado es transitorio, cuando la intensidad en dicha feromona sea 0 el estado de la celda volverá a ser 2.
- $c(i, j) = 1$  : indica una celda que contiene feromona atrayente, este estado es invariante en el tiempo.
- $meta(k)$  : es un número entero asociado a cada agente, cuyo valor puede ser:
  - 0: indica que se encuentra en modo de búsqueda del objetivo.
  - 1: indica que el agente ya ha encontrado el objetivo o un camino hacia este, por lo que se encuentra distribuyendo la feromona atrayente o buscando el camino final al objetivo.
- $pasos$  : representa la cantidad de pasos que debe devolverse cada agente a la hora de comunicar el camino.
- $pasosBot(k)$  : indica la cantidad de pasos que ha dado el agente desde que encontró el objetivo o un camino hacia él.
- $dirBot$  es una función que indica la próxima posición del robot, de acuerdo al algoritmo de movimiento rectilíneo

---

**Algoritmo 4 COFAS:** Comunicación de un objetivo usando feromonas atrayentes

---

**Entrada:** Espacio celular:  $C = J \times J$

Enjambre de  $n$  agentes:  $a$

**Salida:** Cantidad  $x$  de agente que han llegado al objetivo

```
1: for  $k = 1$  hasta  $n$  do
2:    $(i, j) = \text{pos}(a(k))$ 
3:   if  $c(i, j) == 3$  y  $\text{meta}(k) == 1$  then
4:      $x++$ 
5:     break
6:   else if  $(c(i, j) == 3$  y  $\text{meta}(k) == 0)$  o  $(c(i, j) == 1$  y  $\text{meta}(k) == 0)$  then
7:     if  $\text{pasosBot}(k) \neq \text{pasos}$  then
8:        $\text{meta}(k) = 1$ 
9:        $c(\text{pos}(a(k))) = 1$ 
10:       $\text{pos}^{t+1}(a(k)) = \text{dirBot}(a(k))$ 
11:       $\text{pasosBot}(k)++$ 
12:     else
13:        $V = \text{vecinos}(c(i, j))$ 
14:        $m = \text{rand}(V(l))$ 
15:       while  $c(m) \neq 1$  do
16:          $m = \text{rand}(V(l))$ 
17:       end while
18:        $\text{pos}^{t+1}(a(k)) = m$ 
19:     end if
20:   end if
21: end for
22: if  $x \neq n$  then
23:   COFAS( $C, a$ )
24: end if
25: return  $x$ 
```

---

## 5.2. Pruebas experimentales del algoritmo COFAS

Con base al algoritmo diseñado en la sección anterior, se realizaron pruebas para comprobar su eficacia comparándolo con un algoritmo donde no exista comunicación para dar con un camino hacia el objetivo. Además se utilizaron como parámetros iniciales:

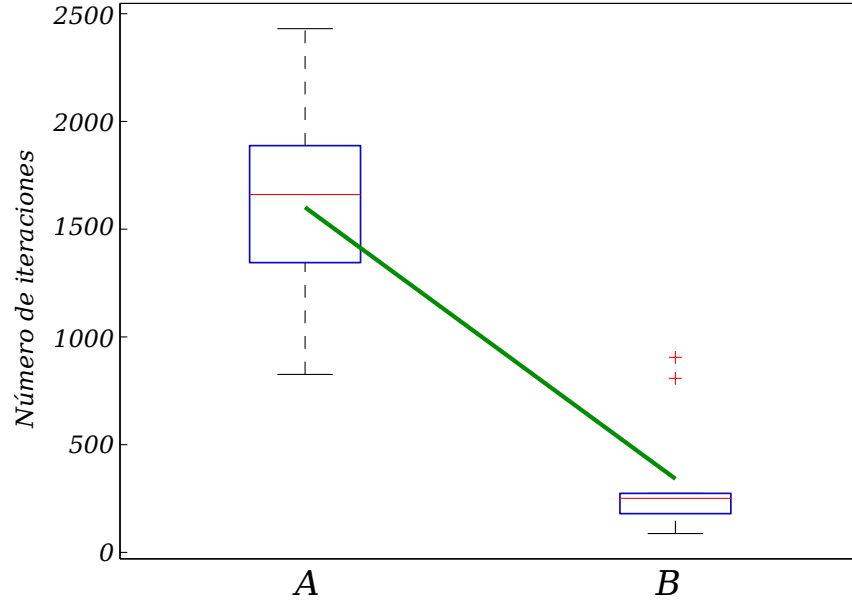
- **Algoritmo base:** Se utilizó el algoritmo confeccionado en la sección 4.4 como punto de referencia para que los agentes buscaran el objetivo. Además este mismo algoritmo sirvió como punto de comparación cuando no se utilizaba comunicación del camino encontrado hacia la meta.



- **Cantidad de iteraciones:** se refiere al número de “pasos” necesarios por el último agente del enjambre que llegó al objetivo.
  
- **Cantidad de agentes en el enjambre e intensidad de la feromona repelente:** se utilizaron 10 agentes en el enjambre y una intensidad inicial de 700 en la feromona repelente. Esto se decidió debido a que en la sección 4 se mostró que dichos valores generaban un mejor desempeño del algoritmo bas.
  
- **Cantidad de pasos de comunicación:** en esta primera parte de las pruebas, se utilizaron 10 pasos de referencia, para que cada agente se devolviera para comunicar el camino.
  
- **Cantidad de repeticiones:** con el fin de obtener datos estadísticos relevantes se realizaron 10 repeticiones de cada uno de los modelos.

Analizando los resultados obtenidos (Ver Figura 30) podemos observar como el algoritmo que utiliza feromonas atrayentes mejora significativa el tiempo que duraría todo el enjambre en encontrar el objetivo sin hacer uso de ningún tipo de comunicación, lo cual es consistente con la teoría existente [13], [28], [29].

Se puede observar como con el algoritmo de comunicación se producen dos datos atípicos, lo cual es normal debido a que los agentes se comportan de manera autónoma y aleatoria.



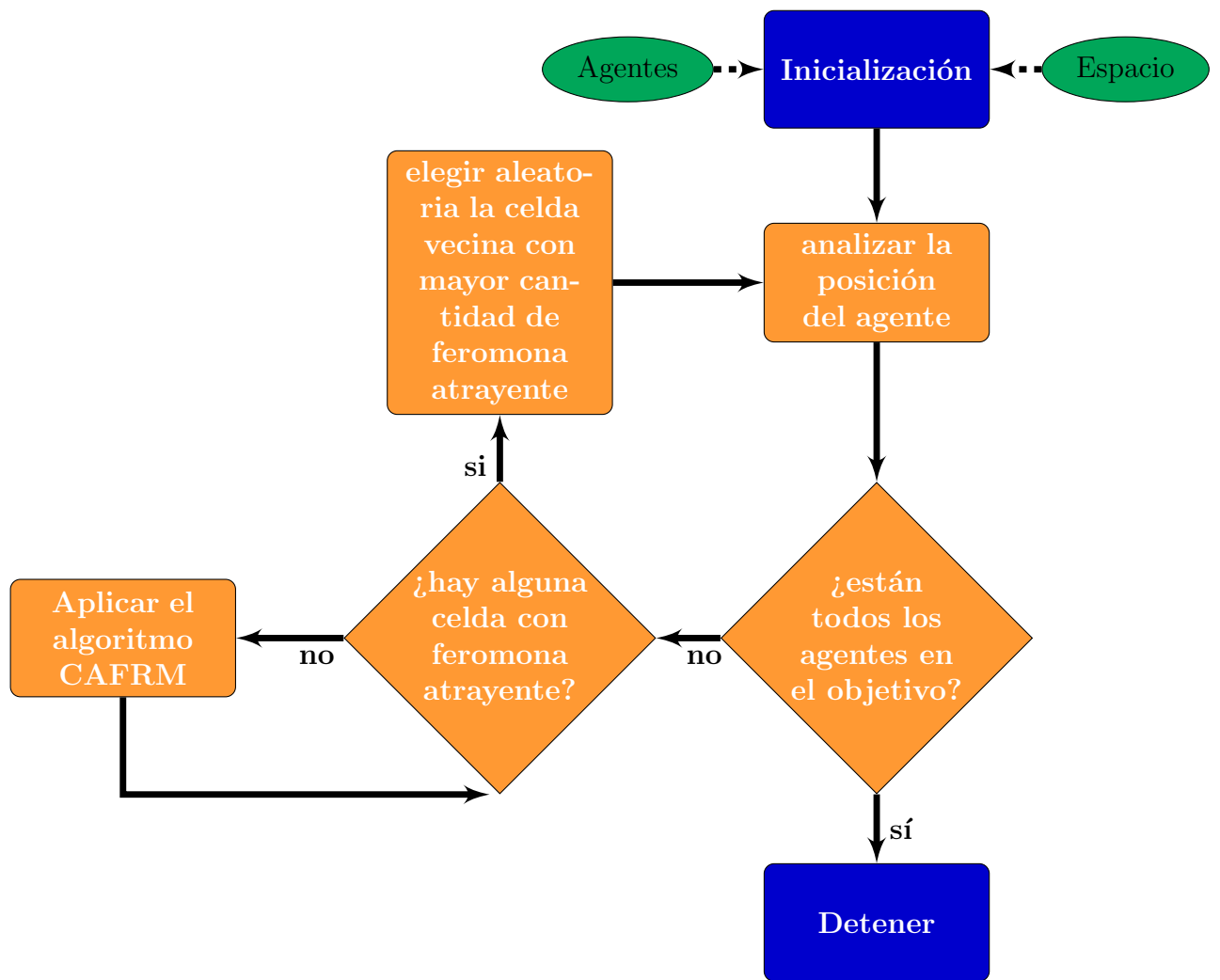
**Figura 30:** Comparación de algoritmos. A. Método sin comunicación. B. Método con comunicación por medio de feromonas atrayentes

Sin embargo, al analizar con detalle la ejecución y animación de cada unas de las repeticiones del método se observa que los agentes tienden a transitar por muchas posibles rutas hasta que logran encontrar el objetivo. Esto produce que el método dure un poco más en converger. Por lo que se plantea mejorar el algoritmo encontrado con el fin de que la comunicación de la ruta hacia el objetivo sea más eficiente.

### 5.3. Algoritmo COFAM: Comunicación de un objetivo usando feromonas atrayentes modificado

Con el fin de mejorar la convergencia del algoritmo 4, se propone agregar un peso o una intensidad asociada a cada celda que contenga una feromona repelente. De esta manera conforme el rastro se vaya alejando del objetivo, su intensidad será menor. Este nuevo esquema permite que, a la hora que los agentes busquen el camino hacia la meta, se dirijan por la ruta que tenga mayor peso o intensidad.

Así cada vez que un agente avance a nueva locación en su proceso de comunicación, disminuirá en 1 la intensidad de la feromona. Así el nuevo esquema se puede observar en la Figura 31.



**Figura 31:** Esquema del algoritmo COFAM: Comunicación de un objetivo usando feromonas atrayentes modificado

El pseudocódigo asociado a dicho esquema se puede observar en el algoritmo 5, en el cual se utilizó la misma simbología con la que se ha venido trabajando.

---

**Algoritmo 5 COFAM:** Comunicación de un objetivo usando feromonas atrayentes modificado

---

**Entrada:** Espacio celular:  $C = J \times J$

Enjambre de  $n$  agentes:  $a$

Cantidad de *pasos* de comunicación

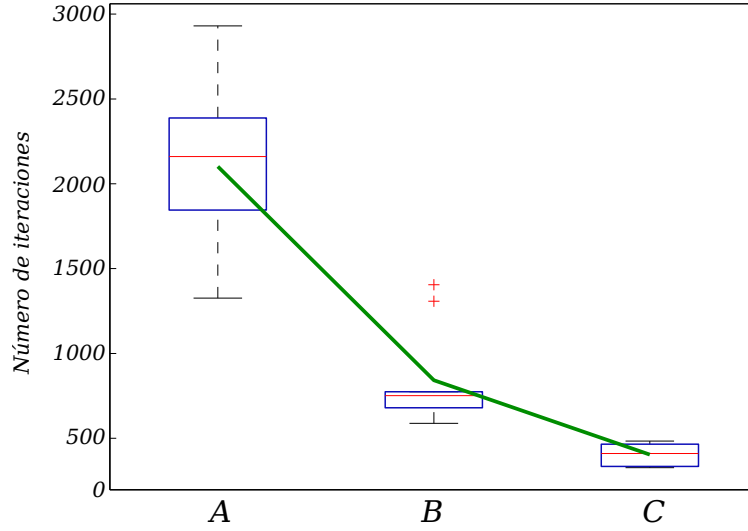
**Salida:** Cantidad  $x$  de agente que han llegado al objetivo

```
1: for  $k = 1$  hasta  $n$  do
2:    $(i, j) = \text{pos}(a(k))$ 
3:   if  $c(i, j) == 3$  y  $\text{meta}(k) == 1$  then
4:      $x++$ 
5:     break
6:   else if  $(c(i, j) == 3$  y  $\text{meta}(k) == 0)$  o  $(c(i, j) == 1$  y  $\text{meta}(k) == 0)$  then
7:     if  $\text{pasosBot}(k) \neq \text{pasos}$  then
8:        $\text{meta}(k) = 1$ 
9:        $c(\text{pos}(a(k))) = 1$ 
10:       $\text{pos}^{t+1}(a(k)) = \text{dirBot}(a(k))$ 
11:       $\text{pasosBot}(k)++$ 
12:       $I(\text{pos}^{t+1}(a(k))) = I(\text{pos}^t(a(k))) - 1$ 
13:    else
14:       $V = \text{vecinos}(c(i, j))$ 
15:       $m = \max(I(V(l)))$ 
16:       $\text{pos}^{t+1}(a(k)) = m$ 
17:    end if
18:  end if
19: end for
20: if  $x \neq n$  then
21:   COFAM( $C, a$ )
22: end if
23: return  $x$ 
```

---

#### 5.4. Análisis de resultados de los algoritmos cooperativos con enjambres de robots para la comunicación de una ruta hacia un objetivo

Con base a este nuevo algoritmo se procedió a realizar varias ejecuciones del mismo, tomando como base los parámetros mencionados en la sección 5.2, los resultados obtenidos se muestran en la Figura 32.

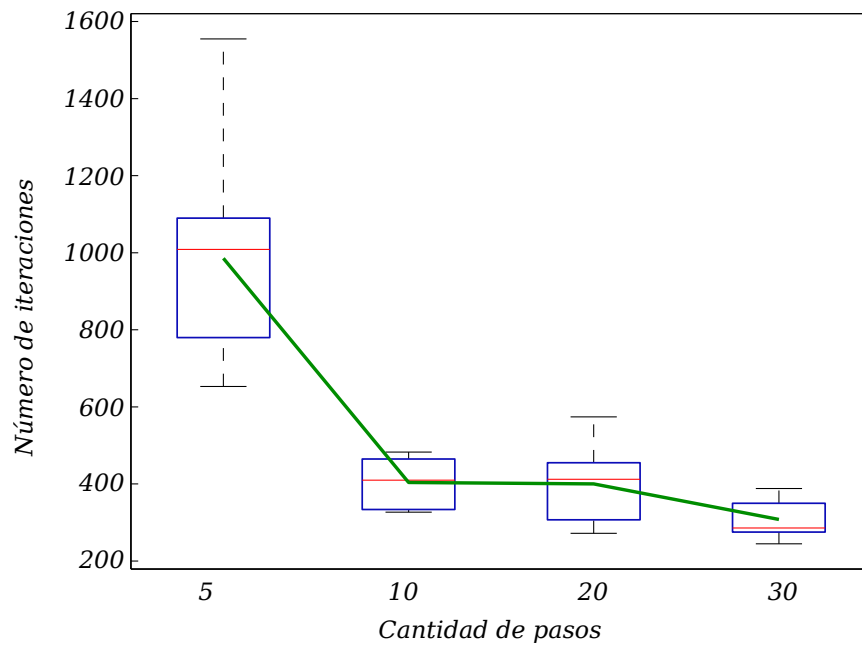


**Figura 32:** Comparación de algoritmos. A. Método sin comunicación. B. Método con comunicación por medio de feromonas atrayentes simple. C. Método modificado con comunicación por medio de feromonas atrayentes

Se puede observar como este algoritmo produce tiempos menores a los otros dos métodos, mejorando significativamente todas las medidas estadísticas empleadas. Además los datos se mantienen en un rango más compacto lo que respalda la consistencia de los resultados, pues al analizar con detención los resultados, observamos que con el algoritmo base CA se obtuvieron tiempos con una desviación estándar de 475.2, con el COFAS un 278.3 y con el COFAM una desviación de solamente 64.

De igual manera, si se analizan los tiempos promedios de los tres algoritmos (CA, COFAS, COFAM) se produjeron tiempos (iteraciones) de 2101, 841 y 403 respectivamente. Lo que evidencia la eficiencia de este algoritmo. Es por esto, que se elije el algoritmo modificado para ser utilizado por el enjambre a la hora de comunicar una ruta.

Cabe recordar que en estas pruebas se utilizó un parámetro de 10 pasos de comunicación, por lo que ahora nos interesa determinar la cantidad óptima de pasos para lograr una convergencia cada vez mayor. Es por esto que se procedió a realizar pruebas con 5, 10, 20 y 30 pasos de comunicación, utilizando los mismos parámetros definidos en la sección 5.2. Los resultados obtenidos se pueden observar en la Figura 33.



**Figura 33:** Resumen estadístico de la ejecución del algoritmo modificado con comunicación por medio de feromonas atrayentes de acuerdo a la cantidad de pasos

Queda claro que conforme la cantidad de pasos aumenta la velocidad de convergencia del algoritmo aumenta y los datos se mantienen cada vez más compactos. Sin embargo conforme se va aumentando las diferencias irán siendo cada vez menores.

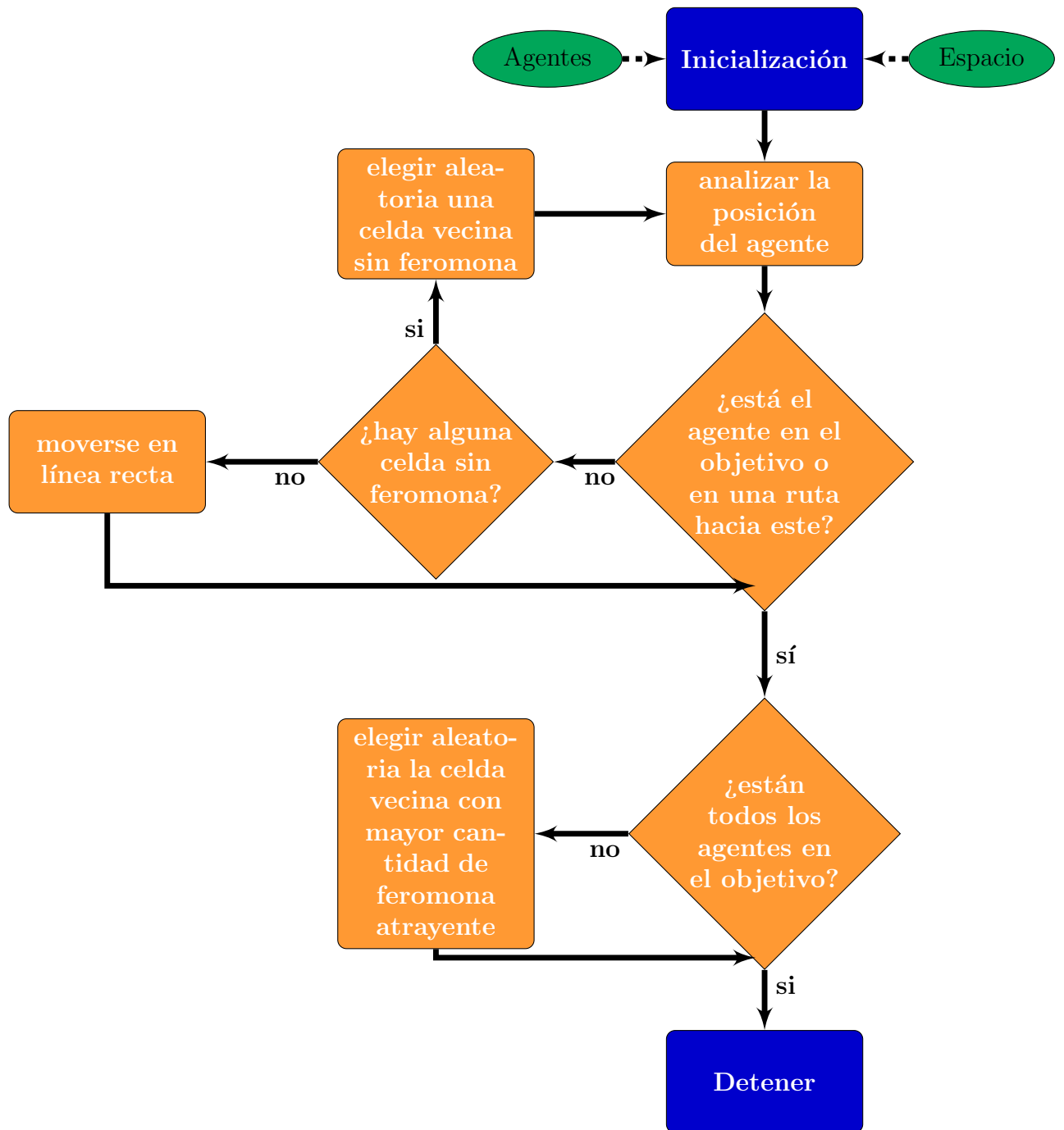
## 6. Algoritmo cooperativo con enjambre de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo

Con base a las pruebas realizadas anteriormente, se propone un algoritmo cooperativo con enjambres de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo. Dicho algoritmo surge de la inspiración biológica que nos ofrecen las sociedades de insectos como las hormigas y termitas, al hacer uso de feromonas para comunicarse con el medio y entre ellas mismas.

Debido a que el algoritmo consiste en dos partes fundamentales y claramente diferenciadas entre sí: búsqueda y comunicación de una ruta, se deben utilizar dos enfoques distintos para conseguir lograr los objetivos de cada uno de ellos. Es por esto que se propone el uso de feromonas de dos tipos:

- **Repelentes:** Las cuales generan en los individuos una repulsión por la zona donde la concentración es mayor, y así logra distribuir de una manera más rápida y efectiva a los individuos por el ambiente.
- **Atrayentes:** Las cuales captan la atención de los individuos, generando así un camino hacia el objetivo.

Al poner en prueba estas ideas se lograron obtener resultados prometedores, por lo que el esquema final está compuesto por la unión de los algoritmos 3 y 5, tal y como se muestra en la Figura 34 y en el algoritmo 6.



**Figura 34:** Esquema del algoritmo AECO: cooperativo con enjambre de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo



---

**Algorithm 6 AEEO:** Algoritmo cooperativo con enjambre de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo

---

**Entrada:** Espacio celular:  $C = J \times J$

Enjambre de  $n$  agentes:  $a$

Cantidad de *pasos* de comunicación

**Salida:** Cantidad  $x$  de agente que han llegado al objetivo

```

1: for cada  $c$  en  $J \times J$  do
2:   if  $c! = 1$  then
3:      $I(c) = \rho \cdot I(c)$ 
4:   end if
5: end for
6: for  $k = 1$  hasta  $n$  do
7:    $V = \text{vecinos}(c(i, j))$ 
8:    $(i, j) = \text{pos}(a(k))$ 
9:   if  $c(i, j) == 3$  y  $\text{meta}(k) == 1$  then
10:     $x++$ 
11:    break
12:   else if  $(c(i, j) == 3$  y  $\text{meta}(k) == 0)$  o  $(c(i, j) == 1$  y  $\text{meta}(k) == 0)$  then
13:     if  $\text{pasosBot}(k)! = \text{pasos}$  then
14:        $\text{meta}(k) = 1$ 
15:        $c(\text{pos}(a(k))) = 1$ 
16:        $\text{pos}^{t+1}(a(k)) = \text{dirBot}(a(k))$ 
17:        $\text{pasosBot}(k)++$ 
18:        $I(\text{pos}^{t+1}(a(k))) = I(\text{pos}^t(a(k))) - 1$ 
19:     else
20:        $m = \max(I(V(l)))$ 
21:        $\text{pos}^{t+1}(a(k)) = m$ 
22:     end if
23:   else
24:     if al menos un  $V(l)! = 0$  then
25:        $m = \text{orden}(V(l))$ 
26:        $\text{minV} = I(m(1))$ 
27:       if  $\text{minV} > \tau$  then
28:          $m = \text{nuevaDir}(V, a)$ 
29:          $\text{pos}^{t+1}(a(k)) = m$ 

```

```

30:     else
31:          $m = rand(V(l))$ 
32:         while  $I(m) \neq minV$  do
33:              $m = rand(V(l))$ 
34:         end while
35:          $pos^{t+1}(a(k)) = m$ 
36:          $I(m) = I_0$ 
37:     end if
38: else
39:      $pos^{t+1}(a(k)) = pos^{t-1}(a(k))$ 
40:      $I(m) = I_0$ 
41: end if
42: end if
43: end for
44: if  $x \neq n$  then
45:     AECO( $C, a$ )
46: end if
47: return  $x$ 

```

---

### 6.1. Análisis de resultados del algoritmo cooperativo con enjambre de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo

Luego de las pruebas realizadas en las secciones anteriores podemos concluir que el algoritmo 6 es el que mejor desempeño presenta, cuando se le compara con otros similares.

Además se logró mostrar que:

1. El esquema de feromonas es ideal cuando se quiere o tiene un enjambre de individuos con poca capacidad computacional y se desea un sistema tolerable a fallos.
2. La intensidad y la tasa de dispersión de la feromona afectan significativamente el rendimiento del sistema. Sin embargo los valores óptimos de éstos dependen del ambiente y la cantidad de individuos que se tengan.

3. Entre más agentes trabajen más rápido convergerá el algoritmo, pero también las diferencias con métodos más sencillos de implementar serán cada vez menores
4. Agregar un peso a una feromona atrayente cuyo objetivo es comunicar una ruta, incrementa sustancialmente la velocidad de convergencia del algoritmo.
5. Entre más pasos de comunicación de un agente más rápido se comunicará la ruta, sin embargo se llegará a un punto donde las diferencias serán cada vez menores. La cantidad óptima depende del entorno en que se esté trabajando.
6. Si se trabaja en un ambiente con zonas muy cerradas (o sea cuando hay habitaciones con una única salida) tiende a haber mucha redundancia de búsqueda en esa sección.
7. El lugar donde sean liberados los agentes o bien desde donde empiecen a sensar el terreno, afecta directamente la convergencia del mismo.
8. En las pruebas realizadas se utilizó un modelo donde los individuos se liberaban todos en una misma zona, la cual es cerrada y está lejos del objetivo. Esto con el fin de analizar los peores escenarios.

## 7. Implementación física de un algoritmo propuesto

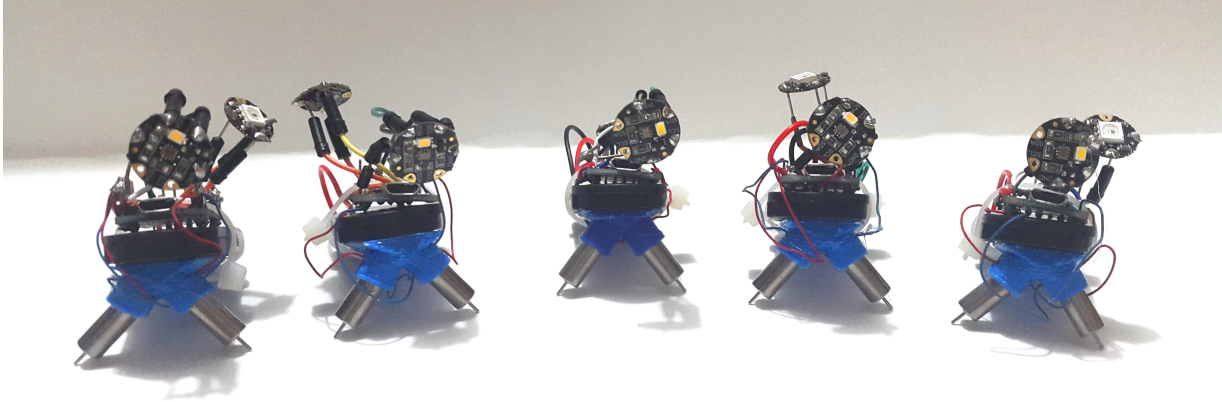
Una vez confeccionado y probado el algoritmo de búsqueda y comunicación de una ruta hacia un objetivo, se procedió a idear la implementación física del algoritmo CA sin comunicación de una ruta hacia un objetivo.

### 7.1. Requerimientos mínimos de hardware para la implementación

Para poder llevar a alguno de los algoritmos presentados en la sección 4.1, se necesita de un robot que sea capaz de:

- Almacenar y ejecutar los algoritmos de búsqueda y comunicación.
- Detectar, por medio de sensores, obstáculos en el ambiente, feromonas tanto repelentes como atrayentes y el objetivo a encontrar.
- Dejar una señal temporal o permanente, según corresponda, en el ambiente; esto con el fin de poder comunicarse con el ambiente y con otros individuos del enjambre.

Con base a estos requerimientos se creó un enjambre de cinco robots haciendo uso de microcontroladores y componentes de bajo costo (ver Figura 35).



**Figura 35:** Enjambre físico de robots

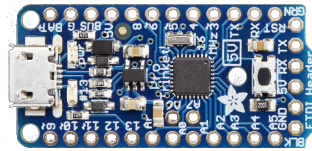
De manera más específica cada uno de estos robots está integrado por los siguientes componentes:

1. **Microcontrolador Pro Trinket 5V:** Es el encargado de controlar los movimientos del robot y la toma de decisiones para poder ejecutar el algoritmo de manera correcta.

Este microcontrolador es creado por Adafruit Industries, LLC y a pesar de su pequeño tamaño (1.5" x 0.7" x 0.2") tiene la misma capacidad que un Arduino UNO (Ver Figura 36). Entre las características principales de este dispositivo tenemos [33]:

- Usa un chip Atmega328P, el cual al ser el mismo que el del Arduino UNO, permite ser programado con el IDE de Arduino, lo cual genera una gran ventaja, pues ese ambiente es altamente similar al IDE de Processing, en el cual se realizaron las animaciones.
- Posee un regulador de 5V que provee 150mA de salida.
- No posee directamente una salida serial, pero se le puede adaptar una mediante un cable FTDI.

- Tiene una velocidad de 16MHz
- El gestor de arranque usa 4KB de Flash por lo que el tamaño máximo del código es de 28.672 bytes. Este gestor no afecta el uso de RAM.
- Es compatible con Linux, Mac OS o Windows.
- Posee un regulador de 5V con 150mA de salida y control de caídas de voltaje. Puede llegar a aceptar entradas de hasta 16V, ofreciendo protección de polaridad inversa, térmica y protección de límite de corriente.



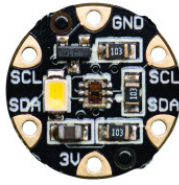
**Figura 36:** Ejemplo de un Pro Trinket 5V (tomado de [33])

- **Sensor de Color TCS34725:** Este sensor, creado también por Adafruit Industries, LLC, permite reconocer el color preciso de un objeto mediante el formato RGB (Ver Figura 37). Posee un filtro de bloqueo integrado que permite reducir al mínimo los componentes espectrales de la luz entrante, lo que permite mediciones de color muchos más precisas. Este sensor también tiene un soporte de circuitos, así, como un regulador de 3.3V.

Este dispositivo es el responsable de sensor el ambiente y determinar, de acuerdo al color, si se encuentra ante:

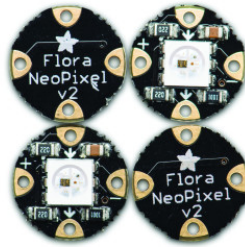
- Espacio libre, determinado por el color negro.
- Obstáculo, determinado por el color blanco.
- Meta u objetivo, determinado por el color amarillo.
- Feromona repelente, determinado por el color rojo.

- Feromona atrayente, determinado por el color verde.



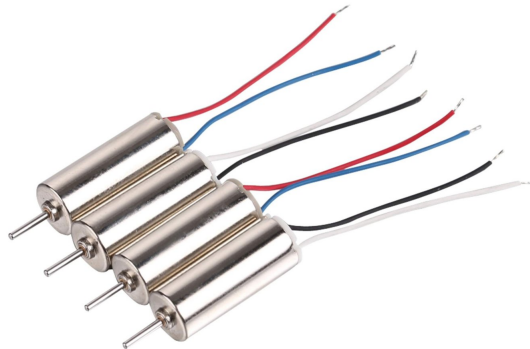
**Figura 37:** Ejemplo de un sensor de color TCS34725 (tomado de [34])

- **NeoPíxeles Flora:** Cada píxel está compuesto por un LED RGB, con una intensidad de 60 mA (Ver Figura 38). Esto permite al robot emitir un haz de luz para ser detectado por otros dispositivos y luego ser proye.



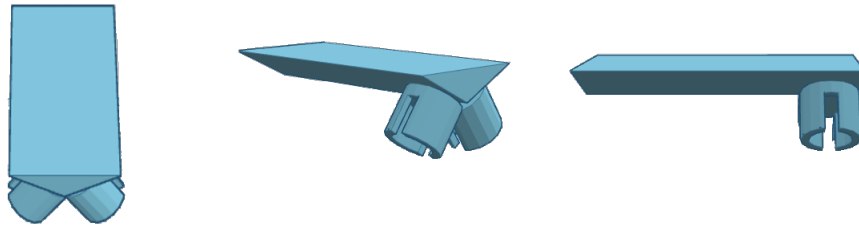
**Figura 38:** Ejemplo de un NeoPixel Flora (tomado de [35])

- **Mini RC Quadcopter Motor:** Este pequeño motor le permite dar movimiento al robot (Ver Figura 39). Cada robot debe contener dos motores y para que su movimiento sea suave y fluido debe estar sobre una superficie lo más prolija posible.



**Figura 39:** Ejemplo de motores mini RC Quadcopter

- **Estructura 3d:** Se creó una estructura en 3d, para que sirviera de base y montura para el ensamblaje de los componentes (Ver Figura 40).

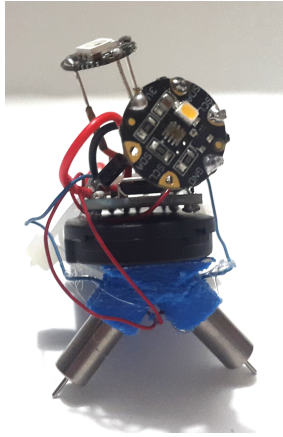


**Figura 40:** Modelado de la estructura 3d para la montura de los robots

- **Otros:** Además de los componentes mencionados anteriormente se utilizaron:
  - Dos baterías CR2032 para proveer un voltaje de 6V.
  - Un portador de dos baterías CR2032.
  - Cables de conexión.
  - Diodos para bajar por seguridad, el voltaje entre el microcontrolador y el sensor de color.

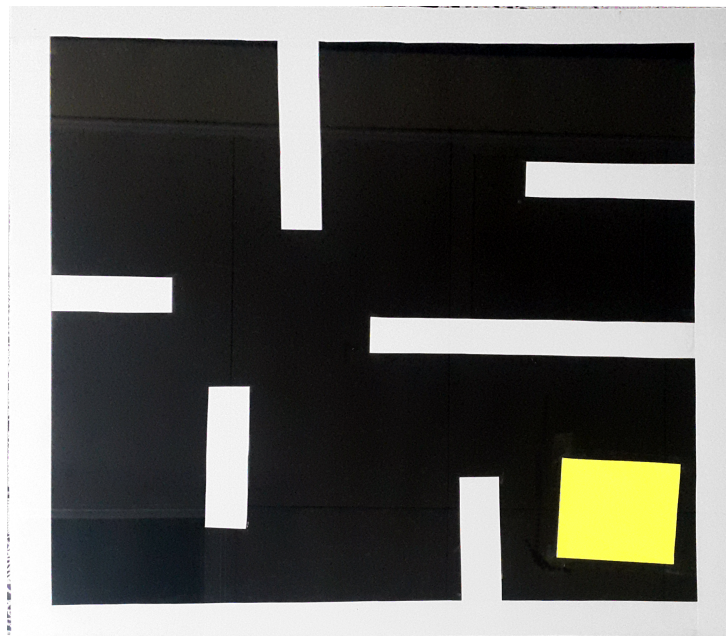
Así el ensamblaje de cada uno de estos componentes permitió la confección de cada uno de los robots (Ver Figura 41).





**Figura 41:** Ejemplo de un robot físico

El ambiente se simuló por medio de un vidrio de 65cm×65cm, al cual se le colocaron materiales que simulan las diferentes partes del entorno (Ver Figura 42).



**Figura 42:** Simulación espacio físico

## 7.2. Análisis de resultados obtenidos por la implementación física

Por cuestiones de complejidad computacional y electrónica, se decidió implementar el algoritmo de caminado aleatorio para la búsqueda de un objetivo sin comunicación de una ruta.

Para esta implementación se utilizó un enjambre compuesto de cinco robots homogéneos en un ambiente estático, tal y como se planteó en la sección anterior.

Los robots se programaron con el software Arduino versión 1.6.9. Cada robot se programó de manera independiente, debido a que se debió calibrar cada uno de los sensores de color y los motores, sin embargo el esquema general para cada uno de ellos se puede observar a continuación:

### Código.

```
#include <Wire.h>
#include "Adafruit_TCS34725.h"
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif

#define PIN 6

#define commonAnode true

Adafruit_TCS34725 tcs =
  Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_50MS, TCS34725_GAIN_1X);
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(1, PIN,
  NEO_GRB + NEO_KHZ800);
```

### Código (continuación).

```
void setup(void) {
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);

  Serial.begin(9600);

  if (tcs.begin()) {
    Serial.println("Found sensor");
  } else {
    Serial.println("No TCS34725 found ... check your connections");
    while (1);
  }
#ifdef __AVR_ATtiny85__
  if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
#endif
  pixels.begin();
}

void loop(void) {
  uint16_t clear, red, green, blue;
  float average, r, g, b;
  tcs.getRawData(&red, &green, &blue, &clear);
  delay(100);

  pixels.setPixelColor(0, pixels.Color(0, 0, 255));
  pixels.show();
  average = (red + green + blue) / 3;

  r = red / average;
  g = green / average;
  b = blue / average;
```

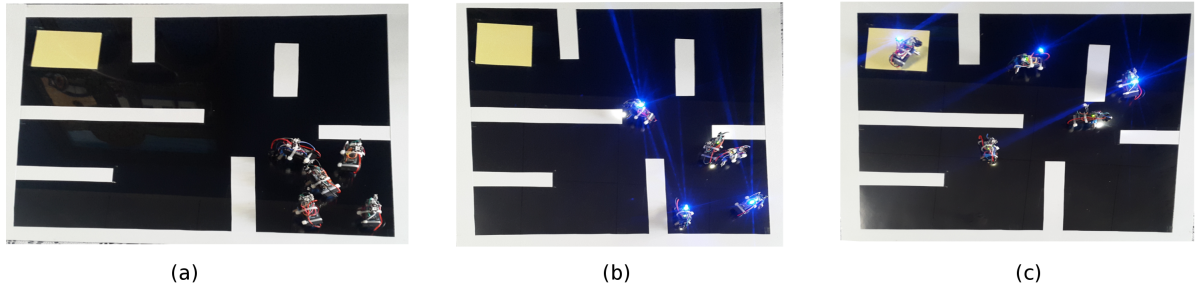
#### Código (continuación).

```
    if ( 0.95<r && r<1.05 && 1<g && g<1.1 && 0.88<b && b<0.99) {
        moveLeft(900);
    } else if (1.15<r && r<1.24&& 1.05<g && g<1.15 && 0.65<b
&& b<0.78) {
        Serial.print("Meta ");
    } else {
        moveLeft(200);
        moveRight(400);
        moveLeft(200);
        moveRight(400);
        moveLeft(200);
        moveRight(400);
    }
}

void moveLeft(int time) {
    digitalWrite(10, HIGH);
    delay(time);
    digitalWrite(10, LOW);
}

void moveRight(int time) {
    digitalWrite(9, HIGH);
    delay(time);
    digitalWrite(9, LOW);
}
}
```

La simulación del espacio físico (ver Figura 42) es una copia a escala del entorno computacional en el cual se realizaron las pruebas de los algoritmos. Por esto se decidió colocar a los cinco robots en posiciones similares a las utilizadas en la simulación (ver Figura 43 parte a).



**Figura 43:** Implementación física del algoritmo. (a) Robots en el punto de salida (b) Robots durante la ejecución del algoritmo (c) Se muestra cuando un robot ha llegado a la meta

Una vez puesto en marcha el algoritmo, este se logró llevar a cabo con éxito (ver Figura 43 partes b y c). Sin embargo se presentaron algunas situaciones que cabe destacar:

- Los robots se mueven de manera muy lenta en el espacio, debido al peso de los mismos y a la capacidad de vibración de los motores.
- Los robots al hacer uso de un sensor de color para detectar obstáculos, tienen a tener unos milisegundos de atraso mientras lo identifican hasta que realizan el cambio de dirección.
- El sensor de color debe estar lo más cerca posible del suelo del ambiente en el que se desarrolla; esto con el fin de determinar de manera más precisa el color de una determinada zona.
- El ruido en el ambiente asociado a factores como: corrientes de aire, polvo, estática, luz en el ambiente y otras que generan interacciones con los sensores y motores que pueden influir en el desarrollo normal de los robots.

## 8. Conclusiones, recomendaciones y trabajo futuro

### 8.1. Conclusiones

En esta investigación se diseñó un algoritmo cooperativo con enjambres de robots para la exploración de una zona y comunicación de una ruta hacia un objetivo. Se logró haciendo uso de un enfoque basado en feromonas e inspirado en el comportamiento biológico de sociedades de insectos como las hormigas y las termitas. Así, se utilizaron feromonas repelentes, para poder dispersar de una manera eficiente y veloz a los individuos de un enjambre con el fin de poner encontrar un cierto objetivo; y feromonas atrayentes para poder comunicar el camino hacia dicho objetivo. Se decidió probar los algoritmos propuestos tanto de manera simulada como de manera física.

Para la implementación computación se optó por trabajar bajo un modelo de autómatas celulares para simular el ambiente y los cambios de estado que ocurren en este. Se utilizó el software Processing, el cual corresponde a un entorno de desarrollo de procesamiento (PDE) y que permite integrar de manera eficiente y practica lenguajes de programación de alto nivel como Java con librerías especiales para simular situaciones físicas y biológicas.

El diseño del algoritmo se dividió en dos secciones:

1. **Exploración de una zona y búsqueda de un objetivo:** En esta sección se utilizaron las feromonas repelentes y luego de las pruebas realizadas se concluyó que:
  - La intensidad y la tasa de dispersión de la feromona afectan significativamente el rendimiento del sistema, por lo cual es necesario encontrar los valores que minimizan el tiempo de búsqueda. Sin embargo estos valores dependen del ambiente y la cantidad de individuos que se tengan.
  - Se puede tener la idea de que entre más intensa sea la señal de la feromona más efectivo será el método. Sin embargo con base a los resultados, se mostró que en este ejemplo en específico, se llega a un nivel que genera el tiempo mínimo y entre más se distancie la intensidad de dicho valor, mayores serán

los tiempos de búsqueda.

- Entre más agentes trabajen más rápido convergerá el algoritmo, pero también las diferencias con métodos más sencillos de implementar serán cada vez menores.
- El lugar donde sean liberados los agentes dentro del ambiente afecta directamente la convergencia del mismo. Entre más distribuidos estén más rápido encontrarán el objetivo propuesto.

2. **Comunicación de una ruta hacia un objetivo:** Para lograrlo se utilizaron, como se mencionó anteriormente, feromonas atrayentes. Al realizar las pruebas respectivas se concluyó que:

- Debe haber algún tipo de comunicación para informar al enjambre que se ha encontrado el objetivo, pues de lo contrario el método converge de manera muy pobre.
- Agregar un peso a la feromona atrayente incrementa sustancialmente la velocidad de convergencia del algoritmo.
- Entre más pasos de comunicación de un agente más rápido se comunicará la ruta, sin embargo se llegará a un punto donde las diferencias serán cada vez menores. La cantidad óptima depende del entorno en que se esté trabajando.

Cabe destacar que como los agentes siguen un esquema relativamente aleatorio, se está propenso a obtener datos atípicos dentro de la ejecución de los algoritmos.

Para las pruebas de la simulación, se utilizó como parámetro de comparación principal el número de “pasos” o iteraciones de cada uno de los agentes. Esto para lograr obtener mediciones más exactas que no se vean afectadas por factores como:

- Capacidad computacional del equipo en que se trabaja.

- Carga de procesador en el momento de la simulación.

La unión de ambas partes del algoritmo se realizó de manera muy natural, gracias en parte, a que ambos utilizan un esquema similar (con feromonas) y a que al utilizar autómatas celulares las reglas asociadas a estos se complementan de manera eficiente.

Se observó que existe una dificultad real a la hora de pasar de la simulación a la implementación física, dado a factores externos que afectan el rendimiento de los robots.

La implementación física debe realizarse en un ambiente lo más controlado posible, donde haya muy poca fricción entre el robot y la superficie en la que se desenvuelve.

Se pueden utilizar muchos componentes, sensores y otros con el fin de detectar de mejor manera obstáculos y huellas virtuales en el ambiente. sin embargo se eligieron aquellas con las características mínimas necesarias para crear un prototipo funcional.

La implementación física trae consigo nuevos retos, por ejemplo:

- Costo económico: pues para poder realizar el algoritmo de la manera más fiel posible, se necesitan varios sensores, motores y otros componentes especiales, que permitan al robot trasladarse por cualquier tipo de ambiente.
- Costo computacional: Los microcontroladores utilizados tienen relativamente poca capacidad de memoria, por lo que la programación debe ser diseñada de manera eficiente con el fin de poder insertarla de manera adecuada. Esto implica que si se tiene un algoritmo extenso, debe optimizarse al máximo para poder ser ejecutado de manera correcta.
- Factores externos: al trabajar con robots físicos se tiene que tener en consideración diversas situaciones que pueden generar ruido durante la ejecución del algoritmo. Ejemplos de esto tenemos:
  - Caídas en el voltaje: ya sea a problemas con la batería o a las conexiones realizadas.



- Problemas físicos con los sensores: lo que puede ocasionar que no lean la información del ambiente de manera adecuada.
- Ruido externo: circunstancias como corrientes de aire, polvo, electrostática, movimientos involuntarios y otros, pueden afectar el desempeño del robot y los sensores asociados a este.

Y aunque sea complicado pasar de la implementación computacional a la implementación física, es necesario ir realizando esfuerzos en esta área; pues es ésta la que permitirá que en un futuro se utilicen dichos algoritmos en escenarios reales donde la búsqueda y comunicación de rutas es una solución a un problema relevante.

## 8.2. Recomendaciones

A partir de las conclusiones anteriores, y para trabajos futuros relacionados con el tema se recomienda:

- Considerar el uso de los enjambres de robots para lograr tareas que, de lo contrario, serían muy complejas de llevar a cabo por un único robots.
- Buscar inspiración en los procesos biológicos que llevan acabo sociedades de insectos, animales y otros; pues éstos son una gran fuente de ideas para lograr resolver problemas relevantes de la vida cotidiana.
- Antes de realizar cualquier implementación física a gran escala, realizar una simulación de la misma para poder analizar la eficiencia de la misma y realizar las mejoras que se consideren necesarias.
- Si se pretende realizar o emular comportamientos de individuos, es preferible emplear un software que integre un entorno de desarrollo de procesamiento, con el fin de realizar las animaciones de manera más sencilla y real.

- Si se quiere realizar la implementación física de un algoritmo, la elección de los microcontroladores y componentes que conforman el robot, debe realizarse con mucho cuidado y tomando en cuenta los alcances de la misma, esto con el fin de identificar los componentes que mejor se adapten al ambiente.

### 8.3. Trabajo futuro

Este trabajo puede extenderse, y pensar en realizar una implementación física completa del algoritmo cooperativo propuesto, con miras a poder aplicarlo en entornos cada vez más reales.

Además los algoritmos programados asumen un mundo discreto continuo sin cambios bruscos en la geografía del ambiente, por lo que se podría extender dichos métodos a ambientes más reales, así se tendría que:

- Realizar una simulación en 3d que tomen en cuenta la geografía propia de un terreno.
- Equipar a los robots de nuevas o mejores partes para que se puedan movilizar de mejor forma por el ambiente.
- Analizar usos de diversas sustancias químicas u otros productos que puedan cumplir el rol de feromonas en el ambiente.

En la ronda de proyectos 2017 que organiza la Vicerrectoría de Investigación y Extensión del ITCR se propuso la puesta en marcha de un proyecto que pretende desarrollar un algoritmo cooperativo centralizado para explorar de manera eficaz el entorno en el que se encuentra un enjambre de robots. Esto con el fin de implementar técnicas de procesamiento y análisis de información, con el fin de lograr construir un mapa de la zona, para luego utilizar optimización multiobjetivo para la determinación de rutas óptimas, de evacuación y/o acceso, utilizando superficies de seudorespuesta y/o enjambre de partículas para disminuir el costo y el tiempo computacional de dicha optimización.

## 9. Bibliografia

- [1] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, “A taxonomy for swarm robots,” in *Intelligent Robots and Systems’ 93, IROS’93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1993, pp. 441–447.
- [2] A. J. Sharkey and N. Sharkey, “The application of swarm intelligence to collective robots,” *Advances in applied artificial intelligence*, p. 157, 2006.
- [3] Y. Tan and Z.-y. Zheng, “Research advance in swarm robotics,” *Defence Technology*, vol. 9, no. 1, pp. 18–39, 2013.
- [4] Y. Mohan and S. Ponnambalam, “An extensive review of research in swarm robotics,” in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE, 2009, pp. 140–145.
- [5] L. L. Grant and G. K. Venayagamoorthy, “Swarm intelligence for collective robotic search,” in *Design and Control of Intelligent Robotic Systems*. Springer, 2009, pp. 29–47.
- [6] B. Lei and W. Li, “A fuzzy behaviours fusion algorithm for mobile robot real-time path planning in unknown environment,” in *Integration Technology, 2007. ICIT’07. IEEE International Conference on*. IEEE, 2007, pp. 173–178.
- [7] J. Tian, M. Gao, and E. Lu, “Dynamic collision avoidance path planning for mobile robot based on multi-sensor data fusion by support vector machine,” in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*. IEEE, 2007, pp. 2779–2783.
- [8] W. Yu, J. Peng, X. Zhang, and K.-c. Lin, “A cooperative path planning algorithm for a multiple mobile robot system in a dynamic environment,” *INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS*, vol. 11, 2014.
- [9] B. Rahnama, M. C. Ozdemir, Y. Kiran, and A. Elçi, “Design and implementation of a novel weighted shortest path algorithm for maze solving robots,” in *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual*. IEEE, 2013, pp. 328–332.
- [10] B. Rahnama, A. Elci, and C. Ozermen, “Design and implementation of cooperative labyrinth discovery algorithms in multi-agent environment,” in *Technological*

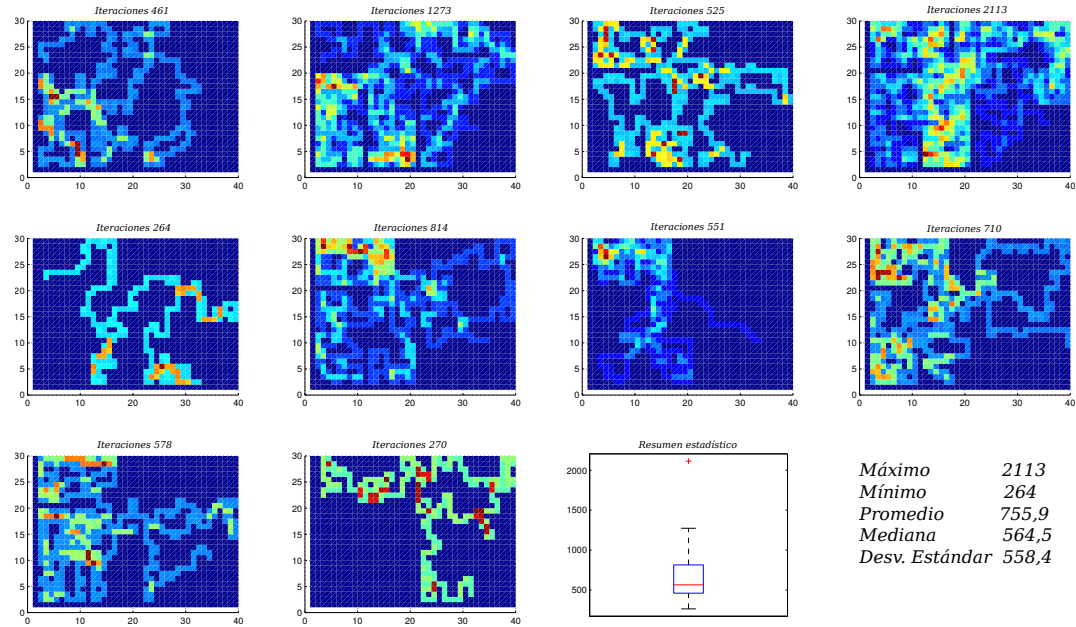
- Advances in Electrical, Electronics and Computer Engineering (TAECE), 2013 International Conference on.* IEEE, 2013, pp. 573–578.
- [11] K. Ioannidis, G. C. Sirakoulis, and I. Andreadis, “A path planning method based on cellular automata for cooperative robots,” *Applied Artificial Intelligence*, vol. 25, no. 8, pp. 721–745, 2011.
  - [12] C. Behring, M. Bracho, M. Castro, and J. Moreno, “An algorithm for robot path planning with cellular automata,” in *Theory and practical issues on cellular automata*. Springer, 2001, pp. 11–19.
  - [13] K. Ioannidis, G. C. Sirakoulis, and I. Andreadis, “Cellular ants: a method to create collision free trajectories for a cooperative robot team,” *Robotics and Autonomous Systems*, vol. 59, no. 2, pp. 113–127, 2011.
  - [14] P. G. Tzionas, A. Thanailakis, and P. G. Tsalides, “Collision-free path planning for a diamond-shaped robot using two-dimensional cellular automata,” *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 2, pp. 237–250, 1997.
  - [15] K. Ioannidis, G. C. Sirakoulis, and I. Andreadis, “Cellular automata-based architecture for cooperative miniature robots.” *J. Cellular Automata*, vol. 8, no. 1-2, pp. 91–111, 2013.
  - [16] F. Marchese, “Cellular automata in robot path planning,” in *Advanced Mobile Robot, 1996., Proceedings of the First Euromicro Workshop on.* IEEE, 1996, pp. 116–125.
  - [17] Y. Tavakoli, H. H. S. Javadi, and S. Adabi, “A cellular automata based algorithm for path planning in multi-agent systems with a common goal,” *International Journal of Computer Science and Network Security*, vol. 8, no. 7, pp. 119–123, 2008.
  - [18] M. Antal, I. Tamas, T. Cioara, I. Anghel, and I. Salomie, “A swarm-based algorithm for optimal spatial coverage of an unknown region,” in *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on.* IEEE, 2013, pp. 7–13.
  - [19] F. Fossum, J.-M. Montanier, and P. C. Haddow, “Repellent pheromones for effective swarm robot search in unknown environments,” in *Swarm Intelligence (SIS), 2014 IEEE Symposium on.* IEEE, 2014, pp. 1–8.

- [20] R. Morlok and M. Gini, “Dispersing robots in an unknown environment,” in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 253–262.
- [21] M. Dorigo, E. Tuci, R. Groß, V. Trianni, T. H. Labella, S. Nouyan, C. Ampatzis, J.-L. Deneubourg, G. Baldassarre, S. Nolfi *et al.*, “The swarm-bots project,” in *Swarm Robotics*. Springer, 2005, pp. 31–44.
- [22] M. Rubenstein, A. Cornejo, and R. Nagpal, “Programmable self-assembly in a thousand-robot swarm,” *Science*, vol. 345, no. 6198, pp. 795–799, 2014.
- [23] M. Dorigo. (2010) The swarmanoid project. [Online]. Available: <http://www.swarmanoid.org/>
- [24] D. Payton, R. Estkowski, and M. Howard. (2005) Pheromone robotics. [Online]. Available: <http://pherobot.webs.com/>
- [25] A. W. Burks, *Essays on cellular automata*. University of Illinois Press, 1970.
- [26] J. Kari, “Theory of cellular automata: A survey,” *Theoretical Computer Science*, vol. 334, no. 1, pp. 3–33, 2005.
- [27] E. F. Codd, *Cellular automata*. Academic Press, 1968.
- [28] F. Arvin, S. Yue, and C. Xiong, “Colias-[phi]: An autonomous micro robot for artificial pheromone communication,” *International Journal of Mechanical Engineering and Robotics Research*, vol. 4, no. 4, p. 349, 2015.
- [29] M. Dorigo, E. Bonabeau, and G. Theraulaz, “Ant algorithms and stigmergy,” *Future Generation Computer Systems*, vol. 16, no. 8, pp. 851–871, 2000.
- [30] J. L. Pearce, B. Powers, C. Hess, P. E. Rybski, S. A. Stoeter, and N. Papanikolaou, “Using virtual pheromones and cameras for dispersing a team of multiple miniature robots,” *Journal of Intelligent and Robotic Systems*, vol. 45, no. 4, pp. 307–321, 2006.
- [31] M. Dorigo. (2005) The swarm-bots project. [Online]. Available: <http://www.swarm-bots.org/>
- [32] B. Fry, C. Reas, D. Shiffman, and et a.l. (2015) Processing. [Online]. Available: <https://processing.org/>
- [33] L. Ada. (2015) Introducing pro trinket. [Online]. Available: <https://learn.adafruit.com/introducing-pro-trinket/overview>

- [34] B. Earl. (2015) Adafruit color sensors. [Online]. Available:  
<https://learn.adafruit.com/adafruit-color-sensors/overview>
- [35] B. Stern. (2015) Sewable neopixels. [Online]. Available:  
<https://learn.adafruit.com/flora-rgb-smart-pixels>

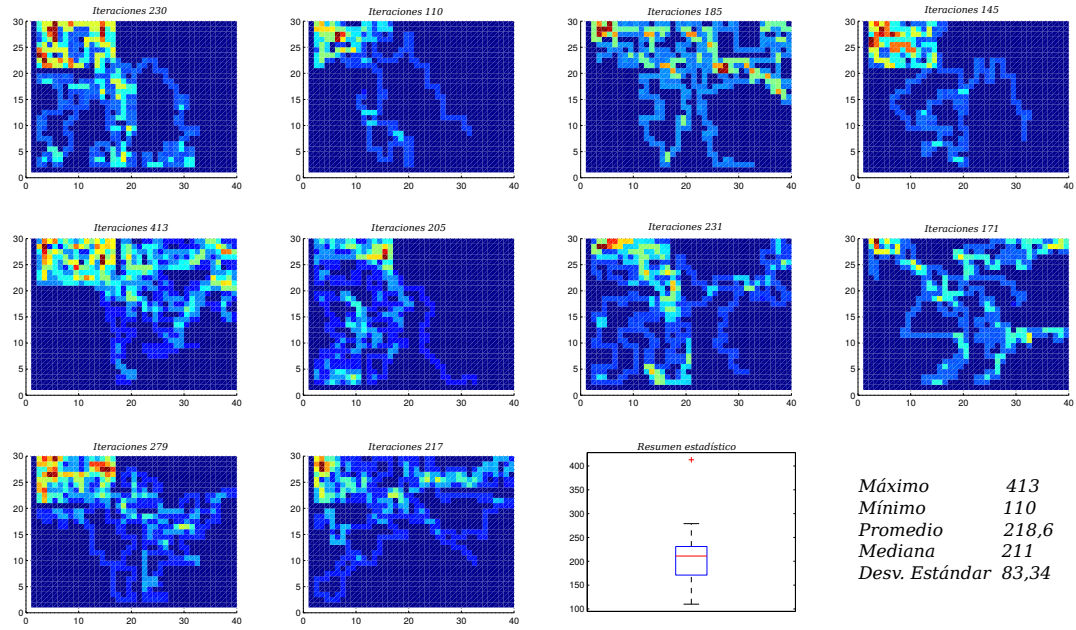
## A. Anexos

### A.1. Resultados del algoritmo de caminado aleatorio con 1 agente



**Figura A.1:** Resultados del algoritmo de caminado aleatorio con 1 agente. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

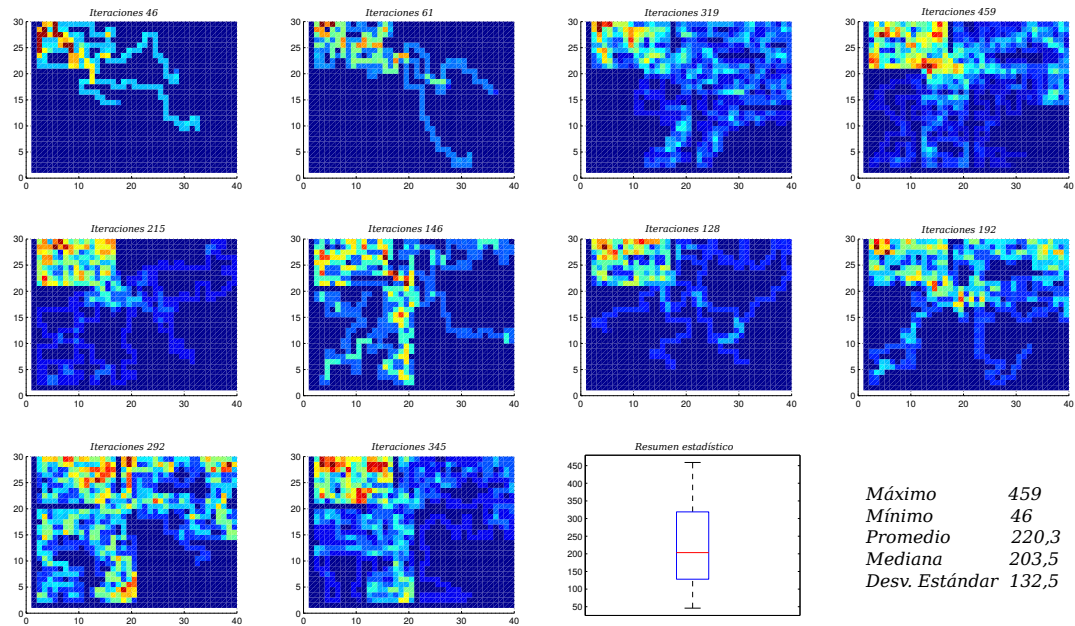
## A.2. Resultados del algoritmo de caminado aleatorio con 3 agentes



**Figura A.2:** Resultados del algoritmo de caminado aleatorio con 3 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

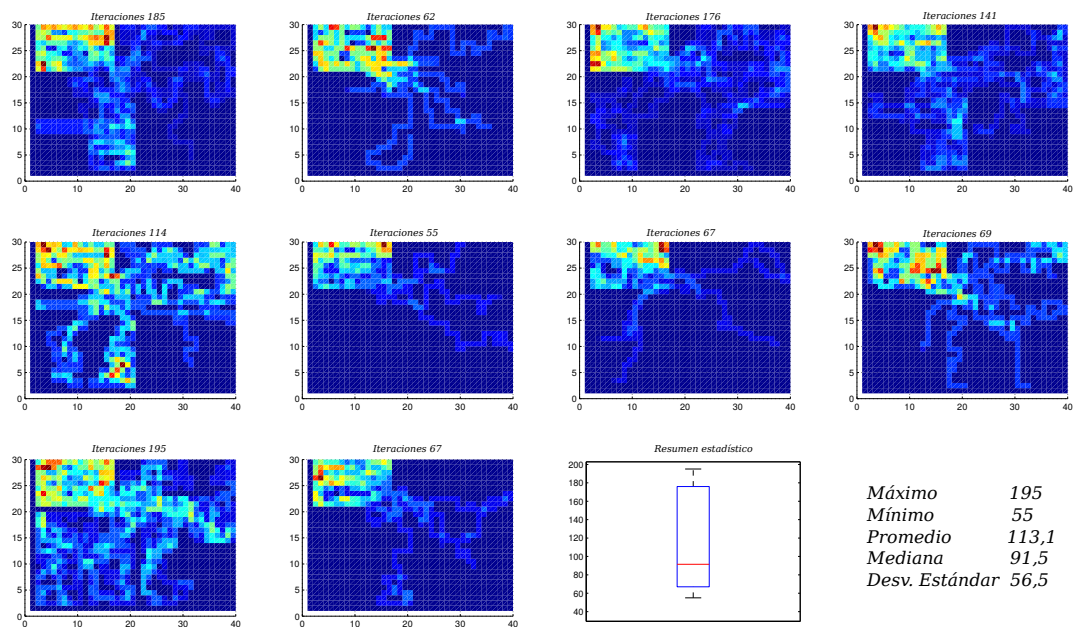


### A.3. Resultados del algoritmo de caminado aleatorio con 5 agentes



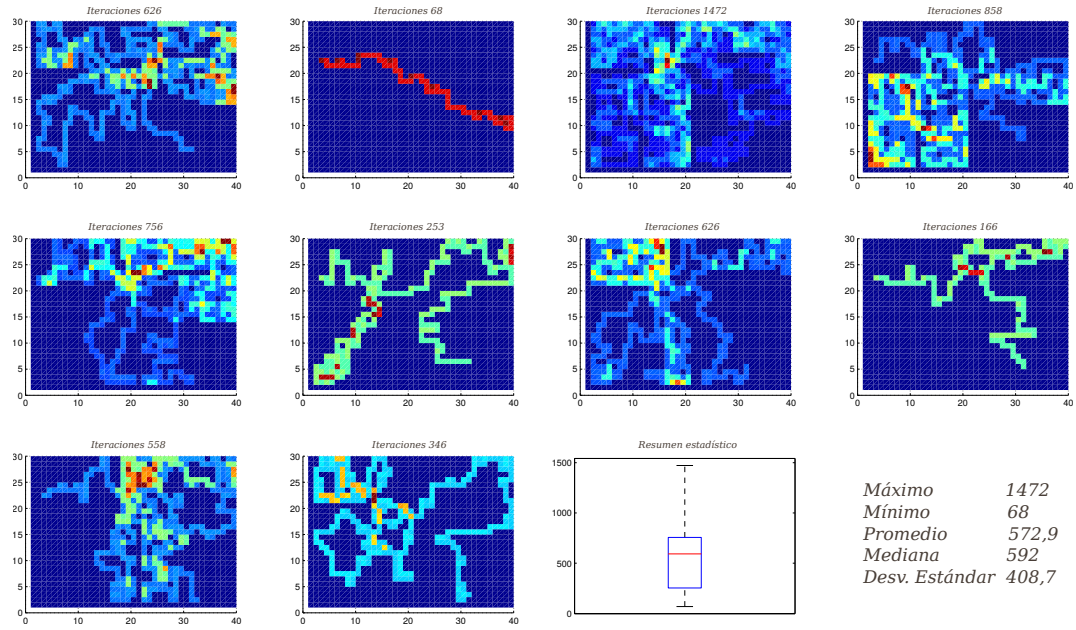
**Figura A.3:** Resultados del algoritmo de caminado aleatorio con 5 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

# A.4. Resultados del algoritmo de caminado aleatorio con 10 agentes



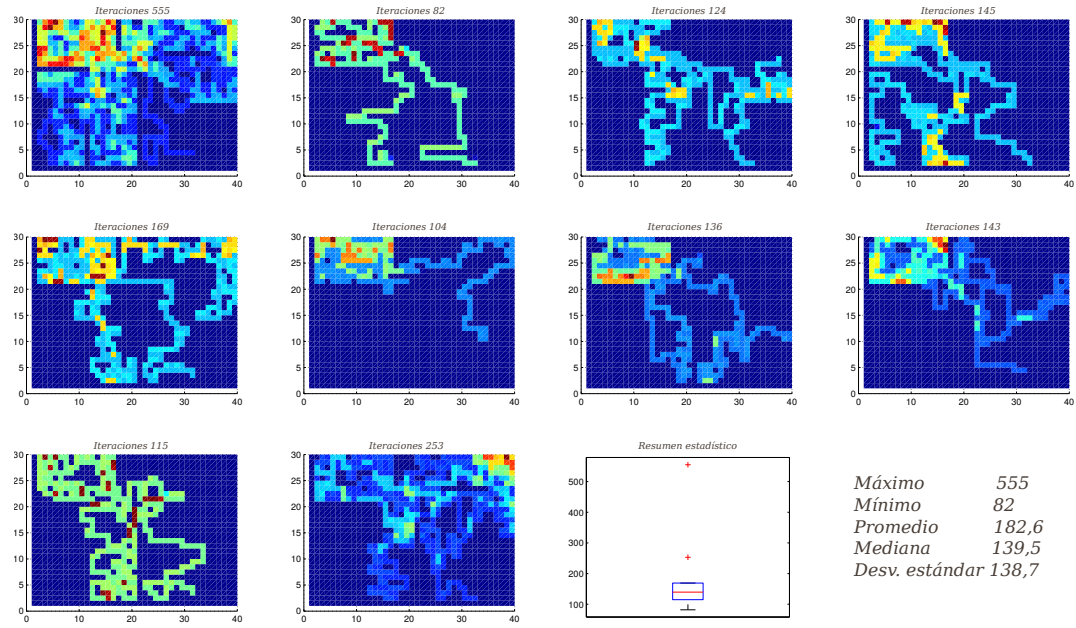
**Figura A.4:** Resultados del algoritmo de caminado aleatorio con 10 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

## A.5. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 1 agente



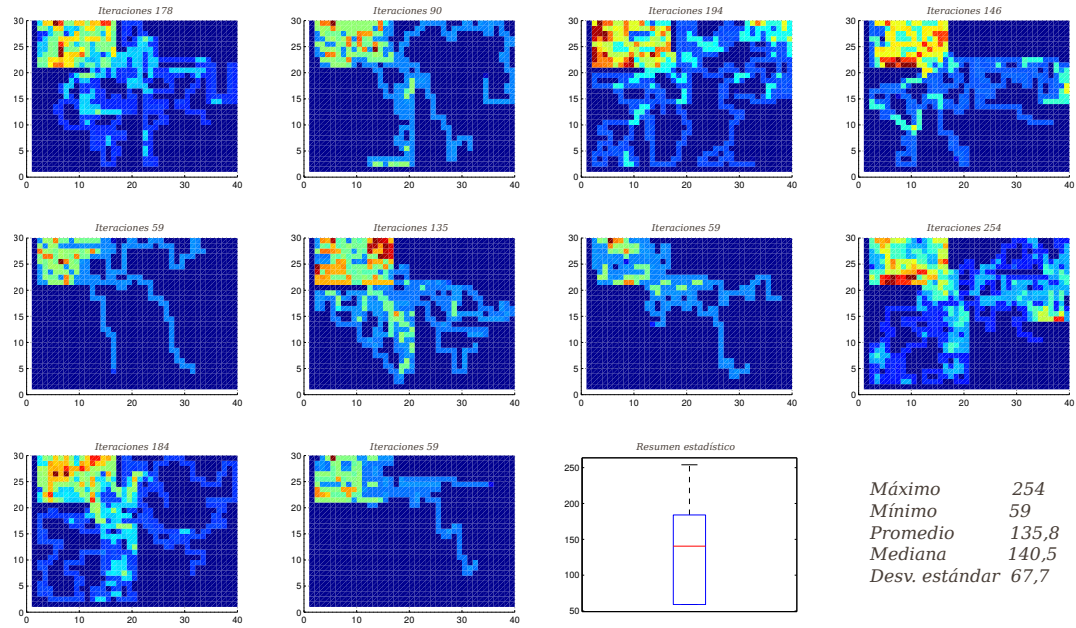
**Figura A.5:** Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 1 agente. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

## A.6. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 3 agentes



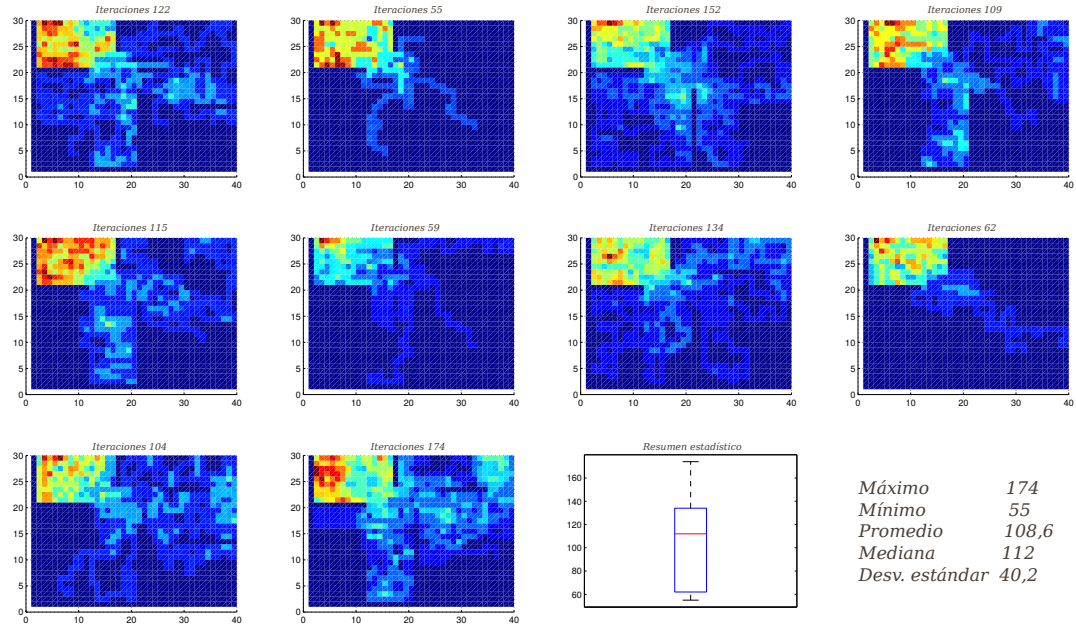
**Figura A.6:** Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 3 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

## A.7. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 5 agentes



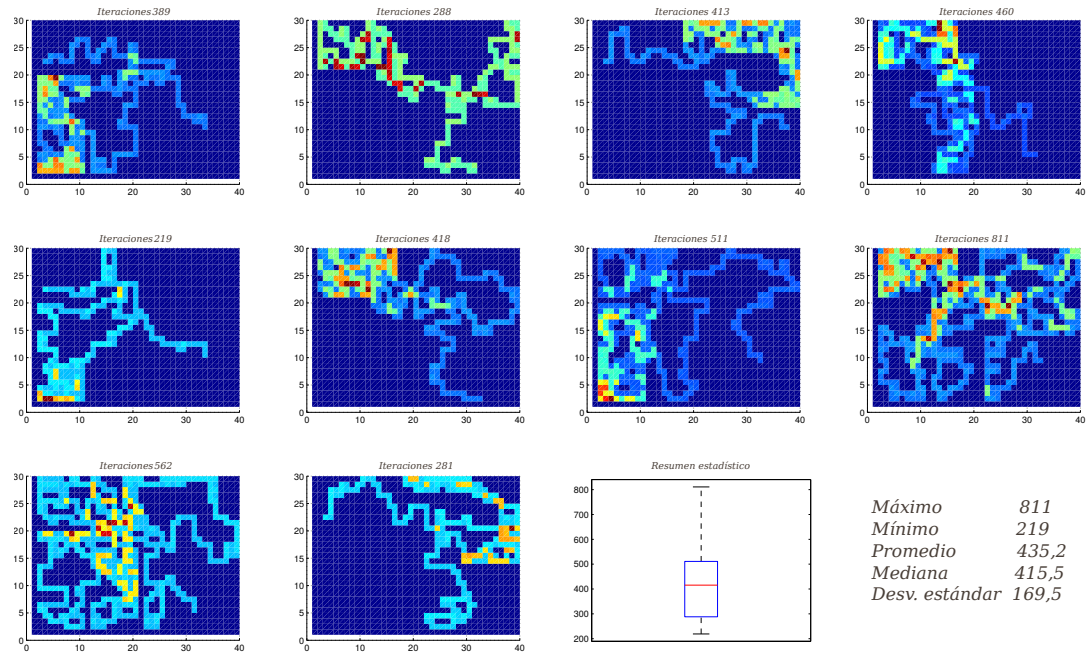
**Figura A.7:** Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 5 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

## A.8. Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 10 agentes



**Figura A.8:** Resultados del algoritmo de caminado aleatorio con feromonas repelentes con 10 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

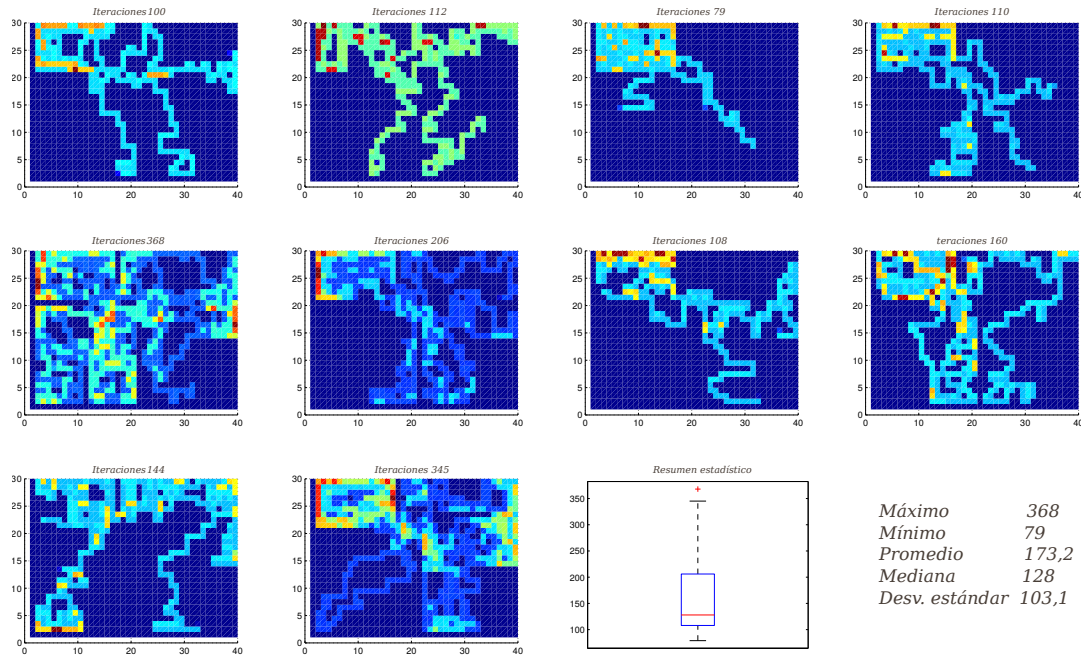
## A.9. Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 1 agente



**Figura A.9:** Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 1 agente. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes



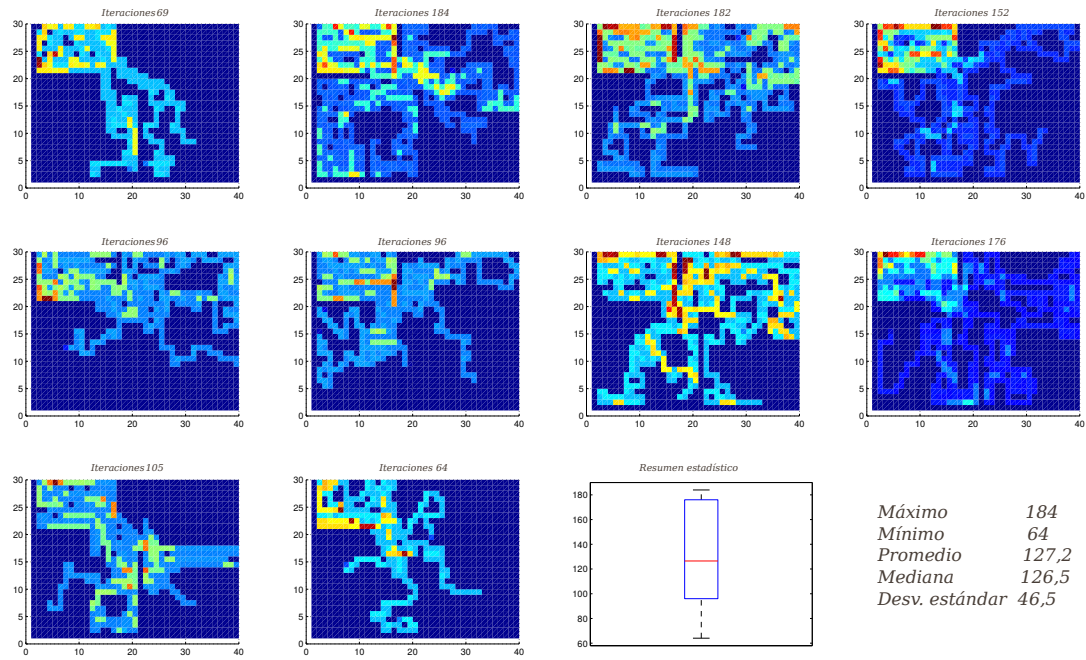
## A.10. Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 3 agentes



**Figura A.10:** Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 3 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

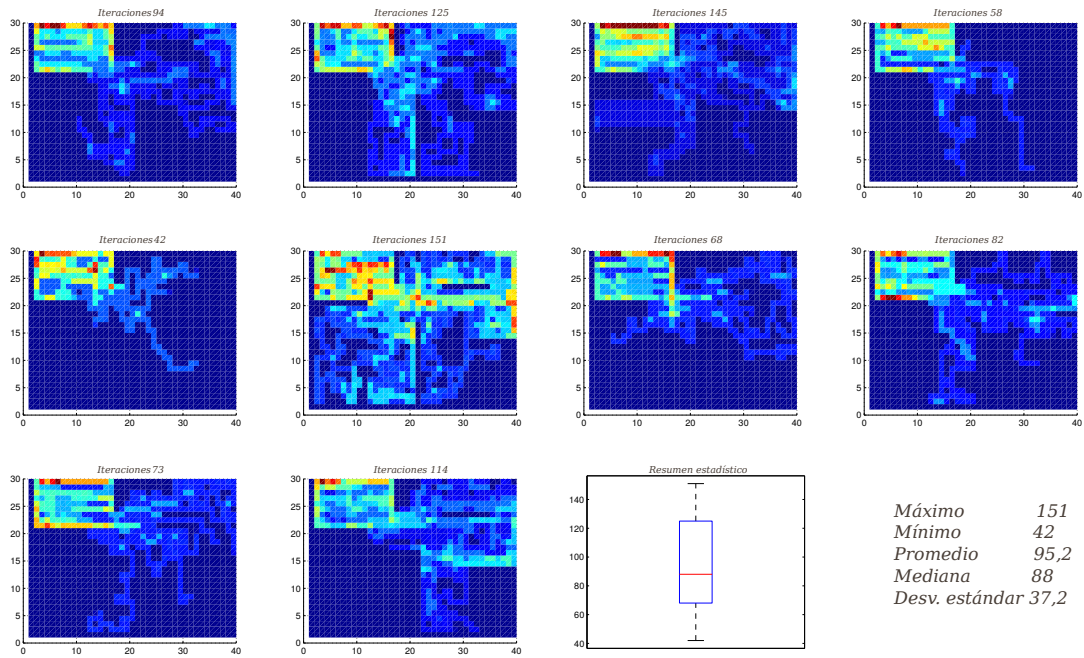


# A.11. Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 5 agentes



**Figura A.11:** Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 5 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes

## A.12. Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 10 agentes



**Figura A.12:** Resultados del algoritmo modificado de caminado aleatorio con feromonas repelentes con 10 agentes. Los colores cálidos indican mayor concurrencia en la trayectoria de los agentes